Dosimétrie in-vivo et contrôle qualité en radiothérapie externe par réseaux de neurones

Frédéric Chatrie Roudier

Université Fédérale
Toulouse Midi-Pyrénées

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ FÉDÉRALE TOULOUSE MIDI-PYRÉNÉES

Délivré par :
l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)

Présentée et soutenue le *17/12/2021* **par :**
Frédéric CHATRIE ROUDIER

Dosimétrie *in-vivo* et contrôle qualité en radiothérapie externe par réseaux de neurones

JURY

Isabelle BERRY	PU-PH, Université Toulouse III	Présidente du Jury
Frédéric ALEXANDRE	DR, CNRS-INRIA de Bordeaux	Rapporteur
Régine GSCHWIND	PU, Université Franche-Comté	Rapportrice
Jean-Luc FEUGEAS	Chercheur CEA, CELIA	Examinateur
Francois HUSSON	Directeur scientifique DOSIsoft	Examinateur
Xavier FRANCERIES	MCF, Université Toulouse III	Co-Directeur
Marie-Véronique LE LANN	PU, INSA de Toulouse	Co-directrice
Norbert BAKALARA	PU, ENSTBB de Bordeaux	Invité

École doctorale et spécialité :
GEETS : Radio-physique et Imagerie Médicale
Unité de Recherche :
Laboratoire d'analyse et d'architecture des systèmes (LAAS-CNRS)
Directeur(s) de Thèse :
Xavier FRANCERIES et *Marie-Véronique LE LANN*
Rapporteurs :
Frédéric ALEXANDRE et *Régine GSCHWIND*

Remerciements

Il me semble très difficile de remercier toutes les personnes qui le méritent, tant elles sont nombreuses à avoir contribué et à m'avoir apporté l'énergie nécessaire pour mener cette thèse à son terme.

Je souhaite commencer par ma directrice de thèse Marie-Véronique Le Lann qui aura tout fait pour que cette thèse se passe dans les meilleures conditions. Elle a toujours été là pour m'aiguiller, me soutenir, puis m'apporter son expérience aux moments souhaités. Une personne très humaine qui donne le maximum pour le bien-être de ses étudiants. Encore merci.

Ensuite, je souhaite remercier mon second directeur de Thèse Xavier Franceries qui m'a permis d'obtenir la bourse de thèse. De plus, je le remercie aussi car il a grandement participer à l'obtention de mon année de césure. Cette dernière m'a apporté la connaissance nécessaire pour mener à bien ces travaux via un Master 2 en Radiophysique Médicale.

M. Frédéric Alexandre et Mme Régine Gschwind m'ont fait l'honneur de bien vouloir être rapporteur de ma thèse, engendrant des discussions constructives et épanouissantes. De plus, leurs remarques m'ont permis d'envisager mon travail sous un autre angle et d'apporter des perspectives à ce dernier. Je les remercie aussi d'avoir accepté cette relecture dans une période surchargée de l'année.

J'aimerais aussi apporter des excuses auprès de M. Olivier Saut pour le retard que j'ai pu avoir sur la remise de mon manuscrit, puisqu'il avait aussi été désigné rapporteur de cette thèse.

Je remercie Mme Isabelle Berry d'avoir bien voulu présider le jury, consacrer du temps à mes travaux de recherche et d'avoir été actrice lors de l'obtention de mon année de césure.

Je remercie également M. Jean-Luc Feugeas et François Husson pour l'honneur qu'ils me font d'être dans mon jury de thèse ainsi que pour l'intérêt qu'ils portent à mes travaux.

Je souhaite aussi remercier tout particulièrement M. Norbert Bakalara pour la curiosité qu'il a pu démontrer envers la recherche effectuée.

Maintenant que j'ai pu remercier les membres du jury, j'aimerais remercier les membres des laboratoires qui m'ont accueilli de manière confortable. Je pense évidemment au chef de l'équipe 15 du Centre de Recherche en Cancérologie de Toulouse, M. Manuël Bardiès, qui a accepté de me fournir un bureau mais aussi le matériel nécessaire pour travailler. De plus, il m'a permis, durant mon année de césure de faire ma première excursion à l'étranger.

Je tiens aussi à apporter des remerciements aux physiciens médicaux, Luc Simon, Laure Parent, Laure Vieillevigne, Régis Ferrand, FX Arnaud, Jocelyne Mazurier et Aurélien Vasseur m'ayant beaucoup apporté de connaissances dans un domaine que je ne connaissais pas. De plus, ils ont grandement participé à la réussite de ces travaux puisqu'ils m'ont permis de faire des mesures et de récupérer des données médicales pour l'application traitée dans ce manuscrit.

Ensuite, je souhaite remercier les deux chefs d'équipes successifs de l'équipe DISCO du LAAS-CNRS Mme Louise Travé-Massuyès et M. Yannick Pencolé pour leurs brillantes idées et explications scientifiques.

Évidemment, je remercie l'intégralité des membres à la fois de l'équipe 15 et de l'équipe DISCO que j'ai pu rencontrer. Un merci particulier pour M. Richard Brown, pour ses idées et points d'intérêts partagés. Je pense notamment aux sessions de sport et moments conviviaux que l'on a pu partager. Je remercie aussi Sara Beilla pour sa gentillesse et ses échanges fructueux. Je la remercie aussi car elle m'a enseignée la radiophysique jusqu'à des heures tardives me permettant d'encadrer correctement et avec plus de recul mes premiers TDs aux PACES. Ensuite, j'adresse mes remerciements à Erick Mora-Ramirez et Gustavo Da Costa pour leur bonne humeur et les échanges culturels riches partagés. Je n'oublierais pas d'adresser mes remerciements à Tony Younes, avec qui on a beaucoup échangé scientifiquement et notamment sur la radiothérapie externe en général.

Mes prochains remerciements s'adressent à M. Maxime Chauvin et Mme Ana Rita Perreira Barbeiro pour qui j'ai une affection particulière. Je vous remercie pour toutes ces discussions instructives à vos côtés, ce déplacement à Montréal et cette gentillesse incarnée que vous portez si bien. Sans oublier M. Jonathan Tranel, je le remercie pour sa philosophie, mais aussi pour sa vision écologique et ces grands débats que l'on a pu avoir.

Je remercie également Édouard Villain et Adrien Dorise, pour les grandes discussions (scientifiques ou non) que l'on a pu avoir et leur bonne humeur dans toutes les circonstances.

Je n'oublierais pas un remerciement particulier à M. Sylvain Durola qui tout au long de mon master a dépensé beaucoup d'énergie dans les enseignements livrés et m'a permis d'évoluer dans le domaine des mathématiques. Je le remercie aussi pour ces entretiens que l'on a pu avoir au début de ma thèse.

Je continue mes remerciements aux personnes les plus importantes dans ma vie, ma famille et mes amis qui se reconnaîtront.

Évidemment, je ne vais pas tous les citer mais il me semble important de commencer à remercier ma femme Mme Élodie Chatrie Roudier avec qui on s'est officiellement unis durant cette thèse et qui m'a intensément soutenue durant ces longues années tant sur le plan personnel que sur le plan professionnel. Je remercie aussi ses parents qui ont corrigés les fautes d'orthographe et qui ont grandement participé à notre bien être durant ces années.

Ensuite, mes remerciements s'adresseront vers mes parents sans qui je n'aurais jamais pu arriver jusqu'ici. Ils ont toujours fait de leur mieux pour me permettre de faire mes études malgré leurs difficultés financières.

Il m'ont permis d'avoir un caractère battant qui me permet de toujours donner le maximum pour arriver à mes fins. De plus, je dédie cette thèse à mon père, qui durant cette dernière a réussi à sortir d'une dégénérescence cérébrale après de multiples batailles. Puis, je remercie mes frères et soeurs qui m'ont soutenu et qui sur cette fin de thèse, sont venus garder ma petite fille Lily, me permettant de considérablement avancer le manuscrit dans ces temps si particuliers avec la COVID'19.

Je finirai mes remerciements qui s'adresseront à ma petite fille Lily qui est arrivée durant cette thèse et qui depuis sa naissance nous apporte sa joie, sa bonne humeur et son enthousiasme d'affronter la vie dans la gaieté.

Table des matières

Remerciements I

Table des matières IV

Acronymes VIII

Glossaire XIII

Introduction générale 1

1 Machine learning et ses applications 5
1.1 Introduction . . . 5
1.2 Techniques de machine learning . . . 5
1.2.1 Types d'apprentissage automatique . . . 7
1.2.1.1 L'apprentissage supervisé . . . 7
1.2.1.2 L'apprentissage non-supervisé . . . 8
1.2.1.3 L'apprentissage semi-supervisé . . . 10
1.2.1.4 L'apprentissage par renforcement . . . 10
1.2.2 Modalités d'apprentissage . . . 10
1.2.2.1 Groupé . . . 10
1.2.2.2 En ligne . . . 11
1.2.2.3 Par transfert . . . 12
1.3 Historique de l'apprentissage automatique . . . 12
1.4 Modèles connexionnistes utilisés . . . 27
1.4.1 Neurone biologique . . . 27
1.4.2 Neurone artificiel . . . 28
1.4.3 Réseaux de neurones artificiels . . . 30
1.4.4 Réseaux de neurones convolutionnels . . . 32
1.4.5 Algorithmes d'optimisation . . . 36
1.4.6 Algorithme de rétropropagation de l'erreur . . . 37
1.4.7 Les algorithmes utilisant la rétropropagation . . . 40
1.4.7.1 Descente de gradient . . . 41
1.4.7.2 Méthode de Newton . . . 41
1.4.7.3 Le gradient conjugué . . . 42
1.4.7.4 Méthode de quasi-Newton . . . 43
1.4.7.5 Méthode de Levenberg-Marquardt . . . 43
1.4.8 L'engouement vers la descente de gradient . . . 44
1.4.8.1 Descente de gradient par lot . . . 44
1.4.8.2 Descente de gradient stochastique . . . 45
1.4.8.3 Descente de gradient par mini-lot . . . 45
1.4.8.4 Optimiseurs spécifiques à la descente de gradient . . . 45

1.4.9 Frameworks existants . . . 45
1.5 Conclusion . . . 46

2 Radiothérapie externe et techniques associées 47
2.1 Introduction . . . 47
2.2 Vue d'ensemble . . . 47
2.2.1 Imagerie . . . 48
2.2.1.1 Tomodensitométrie . . . 49
2.2.1.2 CBCT . . . 50
2.2.1.3 EPID . . . 51
2.2.2 Planification de traitement . . . 51
2.2.3 Dosimétrie . . . 51
2.2.4 Méthode de calcul . . . 53
2.2.4.1 Méthodes analytiques . . . 53
2.2.4.2 Méthodes Monte-Carlo . . . 54
2.2.5 Contrôle qualité . . . 55
2.3 Techniques de traitement . . . 56
2.3.1 Radiothérapie conformationnelle . . . 58
2.3.2 Radiothérapie conformationnelle à modulation d'intensité . . . 59
2.3.3 Arc-thérapie volumique modulée . . . 59
2.4 Conclusion . . . 59

3 L'EPID pour la dosimétrie 61
3.1 Introduction . . . 61
3.2 Notions de dosimétrie . . . 61
3.3 Imagerie portale EPID . . . 64
3.3.1 Propriétés physiques . . . 64
3.3.2 Acquisition . . . 66
3.3.3 Caractéristiques . . . 67
3.3.4 Réponse linéaire en dose absorbée de l'EPID . . . 70
3.4 Dose absorbée de transit et de non transit . . . 70
3.4.1 Approche directe . . . 71
3.4.2 Approche indirecte . . . 72
3.4.3 Approche par réseau de neurones artificiels . . . 73
3.5 Détection d'erreurs à partir de l'EPID . . . 74
3.6 Conclusion . . . 75

4 Contrôle qualité avec les réseaux de neurones 76
4.1 Introduction . . . 76
4.2 Matériels et méthodes . . . 76
4.2.1 Étude des données . . . 77
4.2.1.1 Le format standard DICOM . . . 77
4.2.1.2 Radiothérapie conformationnelle . . . 78
4.2.1.2.1 Machine Varian® . . . 78
4.2.1.2.2 Machine Elekta® . . . 79
4.2.1.3 Radiothérapie conformationnelle à modulation d'intensité 80

4.2.1.3.1 Machine Varian® pour différentes énergies . . 80
4.2.1.3.2 Machine Elekta® 80
4.2.1.4 VMAT 80
4.2.2 Prétraitement des données 81
4.2.3 Modèle et architecture de réseaux de neurones 83
4.2.3.1 Réseaux de neurones feed-forward 85
4.2.3.2 Réseaux de neurones convolutionnels 90
4.2.4 Utilisation du critère clinique gamma 92
4.2.5 Détection d'erreur machine 93
4.3 Résultats et discussions 94
4.3.1 Résultats obtenus à partir des réseaux de neurones classiques . . 95
4.3.1.1 Pour la radiothérapie conformationnelle avec un accélérateur Varian® 95
4.3.1.2 Pour la radiothérapie conformationnelle à modulation d'intensité avec un accélérateur Varian® 96
4.3.1.3 Pour la radiothérapie conformationnelle à modulation d'intensité avec une erreur de traitement simulée . . . 97
4.3.1.4 Pour la radiothérapie conformationnelle à modulation d'intensité avec un accélérateur Varian® et une énergie de 25 MeV 102
4.3.1.5 RC Elekta® 103
4.3.1.6 Elekta® RCMI 104
4.3.2 Résultats obtenus à partir des CNNs 105
4.3.2.1 6 MeV 106
4.3.2.2 25 MeV 109
4.3.3 Analyse et validations des modèles 111
4.3.4 Comparaisons des deux types de modèles 113
4.4 Conclusion 113

5 Dosimétrie *in-vivo* à partir des distances radiologiques 114
5.1 Introduction 114
5.2 Matériels et méthodes 114
5.2.1 Pertinence des distances radiologiques pour ce type de modèle . 115
5.2.2 Modèle mathématiques utilisé 115
5.2.3 Méthodes algorithmiques utilisées 118
5.2.4 CUDA pour l'efficacité du calcul numérique 119
5.2.5 Tests sur différentes cartes (GPU) 125
5.2.6 Encapsulation du code dans Matlab® 125
5.2.7 Prise en compte des hétérogénéités 126
5.2.8 Utilisation des données pertinentes 126
5.2.9 Dosimétrie *in-vivo* 129
5.3 Résultats et discussions 131
5.3.1 Évaluation des performances obtenues avec CUDA 131
5.3.2 Comparaison de l'efficacité entre CUDA, C++ et Matlab® . . . 132
5.3.3 Évaluations des résultats de calcul de dosimétrie *in-vivo* 133

5.4 Pour aller plus loin . . . 136
5.5 Conclusion . . . 138

Conclusion générale **139**

A Fiches techniques fonctions MEX **i**
A.1 Description de l'objectif . . . i
A.2 Forme du fichier MEX . . . i
A.3 Compilation du fichier MEX . . . i
A.4 Utilisation de bibliothèques extérieures . . . ii
A.5 Problèmes de compatibilités rencontrées . . . ii
A.5.1 Windows 32 bits . . . ii
A.5.2 Windows 64 bits . . . iii
A.5.3 Linux . . . iii

Table des figures **iv**

Liste des tableaux **vii**

Liste des Algorithmes **viii**

Bibliographie **ix**

Publications **xxv**
Conférences nationales . . . xxv
Communications orales . . . xxv
Poster . . . xxv
Conférences jeunes scientifiques . . . xxv
Conférences internationales . . . xxvi
Communications orales . . . xxvi
Abstract . . . xxvi
Proceeding . . . xxvi
Posters . . . xxvi
Reviewer . . . xxvii

Acronymes

0D zéro dimension. 3, 72

1D une dimension. 92

2D deux dimensions. vi, xviii, xxiii, 32, 50, 72, 74, 83, 92, 126–129, 137, 140

3D trois dimensions. xviii, xxi–xxiii, xxvi, 2, 4, 48, 50, 51, 59, 61, 72, 73, 88, 92, 115, 118, 119, 127, 136, 137, 140, 141

a-Si Silicium Amorphe. xvii, xix, xx, xxii, xxv, xxvi, 64, 67, 68, 75, 78

AAA Anisotropic Analytical Algorithm. 53, 54

ACP Analyse en Composantes Principales. 9, 13

AdaGrad Adaptive Gradient algorithm. 45

AdaLinE Adaptive Linear Element. 14

Adam Adaptative moment estimation. 45

AE Auto-Encodeur. 18, 21

AN Attention Network. 25

API Application Programming Interface. 122

APIA Applications Pratiques de l'Intelligence Artificielle. xxv

BM Machine de Boltzmann. 17

BTV Biological Target Volume. 49

CapsNet Capsule neural Network. 26

CBCT Cone-Beam Computed Tomography. xxi, V, 48, 50, 51, 55, 72, 137

CCD Charge Coupled Device. 64

CGAN Conditional Generative Adversarial Network. 24

CNN Convolutional Neural Network. iv–vi, xii, xxiii, VI, 8, 16, 18–20, 22–25, 27, 32, 33, 36, 46, 76, 84, 90, 91, 94, 105–111, 113, 130, 140

CNTK microsoft CogNitive ToolKit. 45

CPU Central Processing Unit. 26, 119, 120, 122, 131, 132

CQ Contrôle Qualité. iv, xxv, V, 3, 4, 48, 51, 56, 69, 74, 76–115, 126, 129, 133, 134, 136, 138–141

CT Computed Tomography. vi, xvii, xx, xxiv, 4, 48–52, 59, 69, 71–74, 77, 115–119, 123, 125–129, 131, 136–138, 140

CTV Clinical Target Volume. 49

CUDA Compute Unified Device Architecture. i, xxiv, VI, 4, 114, 115, 119–122, 124–126, 131, 132

DAE Auto-Encodeur Débruiteur. 21

DBN Deep Belief Network. 10, 21

DCGAN Deep Conditional Generative Adversarial Network. 24

DCIGN Deep Convolutional Inverse Graphics Network. 25

DDA Distribution de Dose Absorbée. 2, 3, 50, 52, 53, 57–59, 61, 63, 70, 72–74, 76–87, 89, 92–111, 113–115, 118, 126, 127, 129, 131, 133–135, 137, 139

DF Dark-Field. 69, 70, 78–80, 127

DICOM Digital Imaging and COmmunications in Medicine. V, 66, 67, 73, 77–80, 94

DIV Dosimétrie *In-Vivo*. iv, vi, xxv, VI, 3, 4, 51, 62, 69, 74, 81, 87, 114–138, 140, 141

DL Deep Learning. ix, xiv, xv, xxiii, xxvii, 1, 6, 15–17, 20, 22, 24, 25, 32, 45, 46, 85, 137

DN Deconvolutional Network. 22, 25

DNC Differentiable Neural Computer. 26

DNN Deep Neural Network. xiv, 14–16, 30

DRR Radiographie Digitale Reconstruite. 51, 77

DSD Distance Source-Détecteur. 68, 78–80, 127, 133, 135

DSP Distance Source-Peau. 57, 62

DT Decision Tree. 8, 20

DTA Distance To Agreement. 92

ECMP European Congress of Medical Physics. xxvi

ELU Exponential Linear Unit. 29, 30

EPID Electronic Portal Imaging Device. iv–vi, xviii–xxiii, xxv, xxvi, V, 3, 4, 48, 51, 55, 56, 61–89, 93–111, 115, 118, 126–129, 131, 133–140

ESN Echo State Network. 21

ESTRO European Society for Therapeutic Radiotherapy and Oncology. x, xviii, xxii, xxiv, xxvi

FC Fully Connected. 33

FF Flood-Field. 69, 70, 78–80, 127

FFNN Feed Forward Neural Network. 8, 14, 16–19, 21, 33, 46, 76, 84, 89, 94, 95, 130, 139, 140

GAE Graph Auto-Encodeur. xv, 27

GAN Generative Adversarial Network. iv, xiii, 9, 23, 24

GAT Graph ATtention network. xv, 27

GCNN Graph Convolutional Neural Network. 27

GNN Graph Neural Network. xv, 27

GPC Graphics Processing Clusters. 120

GPS General Problem Solver. 14

GPU Graphics Processing Unit. vii, xxiv, VI, 1, 22, 45, 114, 119–122, 124, 125, 131–133

GRNN Graph Recurrent Neural Network. xv, 27

GRU Gated Recurrent Units. 24

GTV Gross Tumoral Volume. 49

Gy Gray. 2, 47, 55, 57

HN Hopfield Network. iv, 16, 17

IA Intelligence Artificielle. iv, 1, 2, 5, 6, 12–19, 22, 23, 26, 27, 32, 137, 140

IAD Intelligence Artificielle Distribuée. 19

IBM International Business Machines corporation. 20, 22

ICB Institut de Cancérologie de Bourgogne. 64

ICCR International Conference on the use of Computer in Radiation therapy. xxvi

ICRU Commission Internationale des Unités et mesures Radiologiques. xvi, 49

ILSVRC Imagenet Large Scale Visual Recognition Challenge. 23, 26

IMRT Intensity Modulated Radiation Therapy. ix, xviii–xxiv, xxvi, 59

INRIA Institut National de Recherche en Informatique et en Automatique. 46

IoT Internet of Things. 5

IRM Imagerie par Résonance Magnétique. 48, 52

IUCT Institut Universitaire de Cancérologie de Toulouse. 64

IUPESM International Union for Physical and Engineering Sciences in Medicine. xxvi

k-NN k-Nearest Neighbors. 8

KN Kohonen Network. 17

LARD Laboratoires Associés de Radiophysique et de Dosimétrie. xxv

LINAC LINear ACcelerator. iv, xxi, xxvi, 56, 57, 65, 73, 74

LISP LISt Processing. 15

LSM Liquid State Machine. 21

LSTM Long Short-Term Memory. 20, 23, 24

LVQ Linear Vector Quantization. 18

MAE Erreur Moyenne Absolue. 31

MC Monte-Carlo. xvii, V, 53–55, 71–73, 83, 137

MeV MégaélectronVolts. v, vi, VI, 47, 50, 56, 68, 70, 72, 80, 88–90, 94–97, 100–113, 126, 133, 134, 138, 139

MLC Collimateur Multi-Lames. iv, xxi, xxiii, 2, 51, 52, 57, 58, 74, 80, 93, 95, 96, 98, 99, 133, 134

MOSFET Metal-Oxyd Semi-conductor Field Effect Transistor. xviii

MSE Mean Squared Error. 31, 90, 98, 131

MV MégaVolts. 56

Nadam Nesterov adaptative moment estimation. 45

NCR National Cash Register. 18

NTM Machine de Turing Neuronale. 23, 26

NVCC NVidia Cuda Compiler. 122, 123, 126

NVPTX NVidia Parallel Thread eXecution. 122

PCIe Peripheral Component Interconnect express. 120

Penelope PENetration and EnergyLOss of Positrons and Electrons. xvii, 55

PTV Previsional Target Volume. 49

RBFN Radial Basis Function Network. 18

RBM Machine de Boltzmann Restreinte. 10, 17, 21

RC Radiothérapie Conformationnelle. v, vi, V, VI, 2, 58, 59, 76, 78–80, 95, 96, 103–105, 112, 113, 126, 127, 130, 134, 136, 139, 140

RCMI Radiothérapie Conformationnelle à Modulation d'Intensité. v, vi, xvii, xviii, V, VI, 2, 3, 59, 70, 75, 76, 80, 81, 93–97, 100–113, 126, 127, 130, 133–135, 139, 140

ReLU Rectified Linear Unit. 26, 29, 30, 33, 35, 91

ResNet Residual neural Network. 25, 26

RF Random Forest. 8, 20

RMSE Root Mean Squared Error. 31, 91, 107

RMSProp Root Mean Squared Propagation. 45

RNA Réseau de Neurones Artificiels. v, IV, V, 15, 16, 18, 19, 24, 28, 30, 73, 74, 94–97, 100–106

RNN Recurrent Neural Network. iv, 8, 14, 16, 19, 20, 24, 26

SAE Sparse AutoEncoder. 21

SENet Squeeze and Excitation neural Network. 26

SFPM Société Française de Physique Médicale. xxv

SFRP Société Française de RadioProtection. xxv

SFU Special Function Unit. 120, 121

SGD Descente de Gradient Stochastique. IV, 20, 32, 37, 45

SIMT Single Instruction Multiple Threads. 119

SM Streaming Multiprocessor. 120–122, 125

SNN Spiking Neural Network. 21

SoC Security operating Center. 120

SVM Machine à Vecteurs de Support. 8, 20

t-SNE t-distributed Stochastic Neighbor Embedding. 9

TD Temporal Difference. 19

TEP Tomographie par Émission de Positons. 48, 52

TFT Thin Film Transistor. 66

TPC Texture Processor Cluster. 120, 121, 125

TPS Système de Planification de Traitement. 52–55, 59, 62, 63, 66, 70–74, 77–87, 89, 91, 98, 107, 110, 127, 137, 140

UM Unité Moniteur. 52, 55, 57, 62, 67, 68, 70, 82

VAE Auto-Encodeur Variationnel. 22

VGGNet Visual Geometry Group Network. 12, 25

VMAT Arc-Thérapie Volumique Modulée. iv, xix, xxii, xxvi, V, 2, 3, 59–61, 66–68, 75, 80, 81, 137, 138, 141

WGAN Wasserstein Generative Adversarial Network. 24

Glossaire

AlexNet est un CNN proposé par Alex Krizhevsky. 22–25

Inception est un CNN proposé par L'entreprise américaine Google. 24–26

LeNet est un CNN proposé par Yann LeCun. 19, 20, 22, 24

Nvidia est une entrprise américaine qui fabrique principalement des cartes graphiques GPUs. vii, 22, 119, 121, 122, 125

ZFNet est un CNN proposé par Zeiler et Fergus. 23

Introduction générale

Dans les débuts de l'Intelligence Artificielle (IA), les chercheurs abordaient et résolvaient rapidement des problèmes intellectuellement difficiles à résoudre pour les êtres humains, mais relativement simples pour les ordinateurs, autrement dit, des problèmes qui peuvent être décrits par une liste de règles mathématiques formelles. Le vrai défi, de nos jours pour l'IA est de résoudre les tâches faciles à accomplir pour les humains mais difficiles voire impossibles à décrire formellement. Ces dernières années, le domaine de l'IA est en plein essor, avec pléthore d'applications pratiques suivies et financées par un grand nombre d'entreprises et sujets de recherche actifs. Quelques unes des applications pratiques concernent l'automatisation de travail de routine, en passant par la compréhension de la parole, la reconnaissance d'images ou bien même l'aide aux diagnostics en médecine.

Cet essor considérable que l'IA subit, notamment par l'intermédiaire du Deep Learning (DL) [1], est parfois comparé à un ras de marée scientifique, qui est principalement dû aux prouesses technologiques qui amènent des capacités de calculs considérables avec notamment, l'utilisation des cartes graphiques (GPUs). De plus, son développement peut se faire aussi rapidement, grâce au web et à toute l'information partagée par ses utilisateurs. Le web donne accès à tous types de données, textes, images, sons, vidéos, historiques de navigation, capteurs... Toutes ces données requièrent des traitements de plus en plus complexes qui vont au delà de calculs statistiques simples, comme reconnaître des objets ou des personnes dans une image, traduire un texte d'une langue à une autre, rendre la conduite d'un véhicule autonome...

L'un des domaines d'application les plus en vogue concerne la médecine. En effet, les domaines de la biologie, de l'imagerie, de la chirurgie et de la robotique génèrent de plus en plus de données médicales, dont seulement une petite partie est aujourd'hui exploitée. Toute la valeur ajoutée de l'intelligence artificielle va donc reposer sur cette capacité à rassembler les données et les analyser afin d'accélérer la recherche. Déjà un certain nombre d'avancées ont pu être effectuées [2] et nombreuses sont celles qui restent à découvrir.

Cette thèse a été réalisée dans ce cadre puisqu'elle décrit le développement d'algorithmes d'apprentissage automatique dans un objectif lié au traitement de données médicales. Le domaine d'application médical concerne un type de traitement contre le cancer. Cette maladie qui chaque année, tue près de 150 000 personnes en France et 8 millions dans le monde. Les cancers les plus fréquents sont le cancer de la prostate, du poumon et du côlon-rectum chez l'homme et le cancer du sein, du côlon-rectum et du poumon chez la femme. Cette maladie faisant partie des plus meurtrières, après les maladies cardio-vasculaires reste difficile à traiter. La notion de causalité y est pour quelque chose puisque, plus la maladie tarde à être diagnostiquée, plus elle sera difficile à soigner. En effet, la présence de métastases sera plus importante. On parle alors de cancers généralisés. Ceux-ci sont plus difficiles à traiter puisque les cellules cancéreuses

sont localisées à différents endroits. Pour faire face à ces anomalies, plusieurs types de traitements contre le cancer existent et peuvent être classés en trois catégories : la chirurgie, la radiothérapie et les traitements médicamenteux au sens large qui concernent la chimiothérapie, l'immunothérapie, l'hormonothérapie ou des thérapies ciblées. Pour la dernière catégorie citée, on parle de traitement systémique agissant sur le corps dans sa globalité, contrairement à la chirurgie et la radiothérapie, qui sont des traitements loco-régionaux. Deux tiers des traitements font appels à la radiothérapie, ce qui la place comme étant un traitement essentiel. Il existe un éventail de combinaisons de traitements possibles pouvant associer une intervention chirurgicale à la radiothérapie.

La radiothérapie est un champ hétérogène en terme de techniques de traitements et ne peut être parcourue de manière exhaustive. Cependant, les principaux sont : La curiethérapie, qui consiste à venir implanter de manière chirurgicale un objet radioactif scellé venant irradier localement les cellules cancéreuses. La radiothérapie interne vectorisée aussi appelée radiothérapie métabolique qui utilise un produit radiopharmaceutique non scellé qui est injecté, la plupart du temps, par voie intra-veineuse au patient. Ce produit est composé de molécules vectrices et associées au radionucléide que l'on appelle traceur pour sa capacité à transmettre de l'information aux détecteurs en plus de sa capacité à traiter. Enfin, la radiothérapie externe qui fait l'objet de ces travaux. On peut voir la pertinence de lier les domaines de l'IA avec la radiothérapie externe par l'intermédiaire de différents travaux [3-8].

La radiothérapie externe utilise une source radioactive qui est située à l'extérieur du patient. Cette source est créée par l'intermédiaire d'un accélérateur linéaire de particules qui, via son faisceau, va irradier le patient. La quantité d'irradiation reçue par le patient est mesurable et elle est appelée dose absorbée. Son unité est le Gray (Gy). La dose absorbée est élevée sur les volumes cibles (cellules cancéreuses) et minimisée sur les tissus sains environnants. La recherche permet de développer des techniques de plus en plus élaborées dans ce champ de traitement. Il y a quelques années, c'était principalement la Radiothérapie Conformationnelle (RC) qui était utilisée. C'est une technique utilisant un système multi-lames MLC placé dans la tête de l'accélérateur de particules permettant de se conformer à la forme de la tumeur de manière statique. Les lames sont correctement positionnées avant l'irradiation et restent statiques durant le traitement.

Plus tard, la Radiothérapie Conformationnelle à Modulation d'Intensité (RCMI) a vu le jour. Elle permet de se conformer dynamiquement à la tumeur, les lames peuvent être en mouvement durant l'irradiation. Ces dernières années, un nouveau traitement a émergé, l'Arc-Thérapie Volumique Modulée (VMAT), elle est une forme évoluée de la RCMI, car le bras de l'accélérateur peut tourner autour du patient durant l'irradiation. Cela permet d'avoir une meilleure conformation à la tumeur en trois dimensions (3D) et d'avoir des temps de traitement plus rapides.

Cependant les nouveaux traitements étant de plus en plus complexes, la présence d'un grand nombre de paramètres peuvent affecter la mauvaise Distribution de Dose Absorbée (DDA) reçue par le patient. Cela peut malheureusement causer des dommages importants allant jusqu'au décès des patients [9]. Par exemple, l'accident de sur-

irradiation de l'hôpital d'Épinal. Suite à ces faits, depuis 2011, la France a rendu obligatoire une vérification accrue et précise comparant la dose absorbée délivrée et la prescription médicale avec une Dosimétrie *In-Vivo* (DIV). Très rapidement, la procédure mise en place par les cliniques consistait à placer un dosimètre ponctuel, tel qu'une diode ou un dosimètre thermoluminescent sur la peau du patient.

Les inconvénients de cette procédure pour la routine clinique sont sa vérification en 0D qui reste pauvre en information compte tenu de la complexité des champs d'irradiations non homogènes de la RCMI, puis le positionnement compliqué du détecteur pour une séance de VMAT (inadapté avec le mouvement du bras durant l'irradiation). C'est pourquoi, des études se sont portées sur l'utilisation de la dosimétrie résiduelle captée par l'imageur portal Electronic Portal Imaging Device (EPID) situé sous le patient. Dans ce cadre, la dosimétrie de transit a en effet tout son intérêt. Sa haute résolution spatiale, sa large zone de détection, sa reproductibilité, sa répétabilité et aussi l'acquisition de l'information durant l'intégralité du traitement font de lui un candidat pour la DIV. Initialement, l'EPID a été fabriqué afin de vérifier le bon positionnement du patient [10-13]. L'objectif était de remplacer les films radiographiques dédiés à cette tâche. La plus-value était un gain de temps conséquent. Rapidement, les chercheurs ont trouvé un intérêt à l'utiliser à des fins dosimétriques, que ce soit pour le Contrôle Qualité (CQ) ou pour la Dosimétrie *In-Vivo* (DIV) [14, 15].

Les travaux décrits dans ce manuscrit s'inscrivent dans cette thématique puisque l'objectif est de développer des algorithmes d'apprentissage automatique effectuant une conversion du signal EPID vers une DDA dans le patient pendant les séances de prétraitement et durant le traitement.

Ce manuscrit se divise en 5 chapitres dont le résumé figure dans les paragraphes suivants :

— Le premier chapitre porte sur un état de l'art des techniques d'apprentissage automatique existantes à ce jour, accompagnées des principales applications les utilisant. Les différents types et modalités d'apprentissage y sont énoncés tels que l'apprentissage supervisé, non supervisé, semi supervisé ou encore l'apprentissage par renforcement.

— Le second chapitre passe en revue les bases nécessaires à la compréhension de l'application issue de la physique médicale, plus précisément les différentes techniques d'imagerie et de traitement utilisées en radiothérapie externe. Les étapes de prise en charge du patient, le déroulement ainsi que le calcul du plan de traitement seront décrits dans ce chapitre.

— Le troisième chapitre est consacré à l'état de l'art de la dosimétrie *in-vivo*. Il présente différents détecteurs utiles pour la DIV. Une section est consacrée à l'imageur portal EPID montrant ses caractéristiques ainsi que ses propriétés physiques. Une étude de la réponse des EPIDs de différents accélérateurs y est fournie. De plus, les différentes approches de calculs dosimétriques basées sur l'EPID et étudiées ces dernières années via des projets de recherche y sont présentées.

— Le quatrième chapitre expose la plus grande partie de ces travaux. Il concerne

l'étude du CQ en utilisant l'approche par réseaux de neurones. Une étude de la physique a permis d'utiliser un modèle d'apprentissage pertinent. Les différentes architectures étudiées y ont exposées ainsi que leurs comparaisons reposant en particulier sur un critère appelé gamma index largement utilisé en physique médicale.

— Le cinquième chapitre concerne le calcul massif de distances radiologiques. Pièce maîtresse dans le calcul dosimétrique, il est important d'avoir accès à cette information de manière efficace. La voie la plus rapide, de nos jours, concerne le calcul massivement parallèle à la condition près, que les calculs soient indépendants. L'implémentation a été faite avec le langage de programmation Compute Unified Device Architecture (CUDA) avant d'être encapsulée dans Matlab®. Une étude de comparaison d'efficacité entre plusieurs langages a été faite. Ce cinquième chapitre traite aussi de l'extension des algorithmes énoncés dans le chapitre 4, afin de les rendre utilisables pour le calcul de la DIV. Plusieurs informations supplémentaires étaient nécessaires telles que le CT qui donne de l'information sur les hétérogénéités de densité électronique du patient ainsi que la distance radiologique traversée par le faisceau. Pour finir, différentes perspectives intéressantes à prendre en considération dans les travaux futurs sont annoncées. Il y est précisé quelques données pertinentes pour la modélisation 3D avec les algorithmes d'apprentissage automatique. Il est aussi montré la difficulté d'utiliser l'EPID avec acquisition en mode continu, car il contient une quantité d'informations assez pauvres lorsque l'on reconstruit un objet 3D.

Chapitre 1

Machine learning et ses applications

1.1 Introduction

Aujourd'hui, nous vivons dans un monde hyperconnecté dans lequel chaque interaction, allant de l'appel téléphonique à l'affichage d'une page web et bien d'autres applications, s'ajoute un océan de données sans limite. Avec l'arrivée de l'internet des objets (IoT), les voitures, les réfrigérateurs, les vêtements de sport ... génèrent des millions de données supplémentaires chaque jour. Elles peuvent être analysées et traitées afin d'atteindre différents objectifs via les algorithmes d'IA. Le concept de l'Intelligence Artificielle (IA) est de faire penser les machines « comme des humains ».

En d'autres termes, effectuer des tâches telles que raisonner, planifier, apprendre et comprendre notre langage. Bien que personne ne s'attende à l'heure actuelle ni même dans un futur proche, à une équivalence avec l'intelligence humaine, l'IA a des incidences importantes sur nos vies. En effet, l'IA a été conçue pour nous rendre plus productif et nous faciliter le travail. Les algorithmes d'IA ont été utilisés pour moult applications ces dernières années. Un panel de techniques existent et sont utilisées pour plusieurs catégories de réalisation. Dans ce chapitre, un état de l'art est proposé en exposant les différents types d'apprentissage automatique ainsi que leurs modalités. De plus, un ordre chronologique d'apparition des techniques a été fait afin de situer les découvertes dans le temps.

1.2 Techniques de machine learning

L'IA ne date pas d'aujourd'hui, elle a commencé dans le début des années 1950. Depuis, beaucoup de recherches ont été faites la concernant et ce n'est que le commencement. Elle a connu des avancées fulgurantes tout comme des déboires à certaines périodes qui sont souvent appelées "hivers". Sa courbe de représentation des avancées est en dent de scie avec deux principaux creux : un dans le milieu des années 1960 et l'autre dans les années 1990. Ces deux hivers sont intervenus suite à un excès d'enthousiasme sans précédent et sans que la recherche n'obtienne les résultats escomptés. Cependant, nous sommes aujourd'hui dans ce que l'on peut qualifier de ras de marée scientifique concernant l'IA. Nous la retrouvons dans un grand nombre de technologies et l'état de l'art est conséquent.

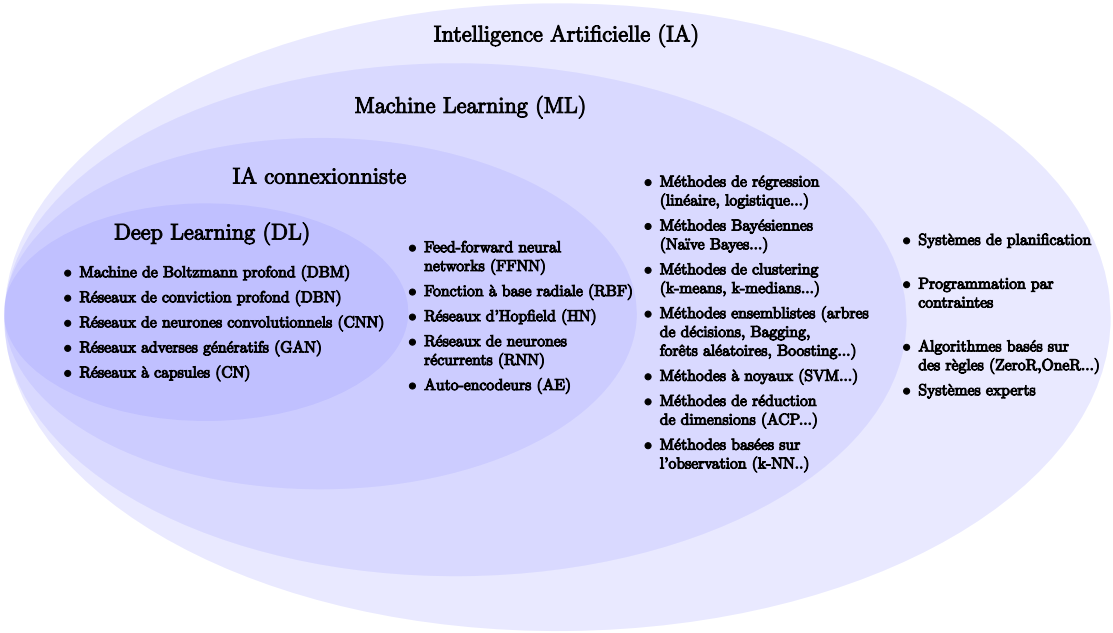


FIGURE 1.1 – Schéma d'ensemble de l'IA

L'IA rassemble de nos jours, un très grand nombre de méthodes. Un échantillon de celles-ci apparait sur la Figure 1.1. Compris dans l'ensemble IA figure le sous-ensemble apprentissage automatique (Machine learning en anglais) qui lui même contient le sous-ensemble IA connexioniste. Pour finir l'ensemble IA connexionniste contient le sous-ensemble apprentissage profond (DLs). Les méthodes exposées dans les différents ensembles sont énumérées de manière chronologique par la suite. Mais avant de revenir sur chacune d'elles, il existe tellement de techniques d'IA qu'elles ont été classées dans différents types et modalités. Toutes ces techniques d'apprentissage automatique fonctionnent, comme les humains avec deux phases bien distinctes : la phase d'apprentissage et la phase d'inférence ou production visibles sur la Figure 1.2. La phase d'apprentissage peut se faire de différentes manières selon le type choisi.

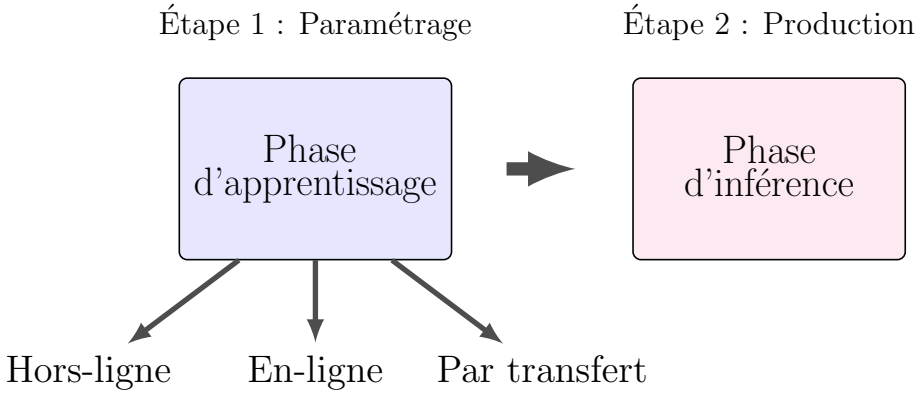


FIGURE 1.2 – Descriptif des phases de l'IA

1.2.1 Types d'apprentissage automatique

Les systèmes d'apprentissage automatique peuvent être classés en fonction de l'importance et de la nature de la supervision qu'ils requièrent durant la phase d'apprentissage. Les algorithmes d'apprentissage devront être choisis en fonction des données fournies, du problème rencontré et des spécificités du modèle à construire. Il existe 4 types majeurs : l'apprentissage supervisé, l'apprentissage non supervisé, l'apprentissage semi-supervisé et l'apprentissage par renforcement.

1.2.1.1 L'apprentissage supervisé

Dans l'apprentissage supervisé, les données d'apprentissage que l'on fournit à l'algorithme comportent les solutions désirées, appelées aussi étiquettes ou cibles selon le problème posé. Par analogie avec l'humain, on parlera d'apprentissage avec un professeur. En effet, le professeur est l'expert qui donne l'explication, ici, l'étiquette correspondant à l'objet. Par exemple, dans le cas d'un classifieur d'images, nous allons avoir un ensemble de données étiquetées.

Ainsi, nous connaissons la correspondance associée à chaque image. À partir de ces étiquettes, nous allons pouvoir apprendre au modèle les différentes catégories représentées. Un autre exemple concerne la régression, un ensemble de valeurs décimales appelées cibles sera fourni. À partir de ces cibles, le modèle va pouvoir "apprendre" les différentes valeurs numériques à associer. Ainsi, pour ces deux exemples, à chaque prédiction du modèle, nous pouvons lui indiquer s'il a donné le résultat attendu ou s'il s'est trompé. De ce fait, le système apprend de ses erreurs. Il cherchera à minimiser l'erreur en fonction des données qu'il possédera. On peut noter que certains algorithmes de régression peuvent être utilisés également en classification et inversement. Par exemple, la régression logistique s'utilise couramment en classification.

Dans certains apprentissages supervisés, la phase d"apprentissage se découpe en 3 parties. Pour cela, l'ensemble des données d'apprentissage est partagé, selon un ratio donné par l'utilisateur, en 3 sous-ensembles. Le premier temps correspond à l'entraînement. Comme son nom l'indique, il permet à l'algorithme de s'entraîner en minimisant l'erreur sur un maximum d'échantillons de données.

La deuxième phase correspond à la validation : il donne accès au comportement du modèle durant sa phase d'entraînement avec des données n'ayant pas été utilisées durant la phase d'entrainement. Il va permettre d'évaluer si le modèle est généralisable avec de nouvelles données, autrement dit, savoir si le modèle n'a pas encouru du sur-apprentissage, connu sous le mot anglais overfitting. Le sur-apprentissage signifie que le modèle a appris des caractéristiques correspondantes à des détails insignifiants dans les données d'entraînement que l'on appelle bruit.

Dès lors, lorsque le modèle est confronté à de nouvelles données, les résultats obtenus ne sont pas ceux escomptés. Pour éviter cela, il existe des techniques de régularisation. Parmi elles, il y a la régularisation L1 (norme 1) ou L2 (norme 2) et plus globalement la norme Ln. La régularisation par norme consiste à ajuster à la baisse la valeur des

poids (paramètres) élevés au sein du modèle proportionnellement à la somme de leurs valeurs absolues pour la norme L1 et à la somme de leurs valeurs au carré pour la norme L2. Un mélange de plusieurs normes avec chacune un ratio d'intervention est possible avec le elastic-net [16]. D'autres techniques de régularisation telles que le dropout [17] ou la normalisation par lot [18] sont souvent utilisées sur les réseaux de neurones. Le dropout consiste à supprimer comme on peut le voir sur la Figure 1.3, à chaque nouvel échantillon, certains neurones forçant le modèle à généraliser. La normalisation par lot est utilisée pour faciliter l'apprentissage et lutter contre des problèmes numériques tels que l'annulation ou l'explosion du gradient.

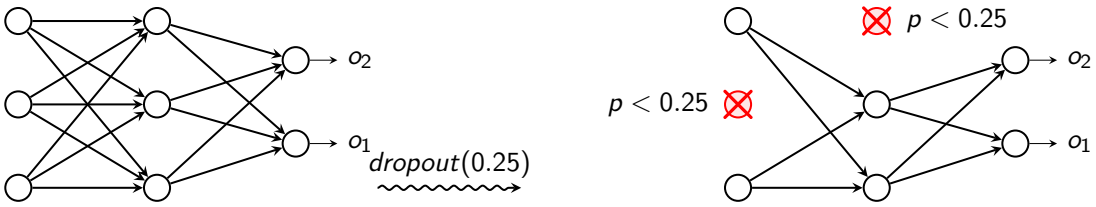


FIGURE 1.3 – Schéma de fonctionnement de la régularisation Dropout

L'explosion du gradient est vue comme l'augmentation très rapide des valeurs des gradients pendant la phase d'apprentissage entraînant un dépassement des limites acceptables de la valeur des nombres par l'ordinateur et donc l'arrêt de l'apprentissage. La validation a une grande importance dans la phase d'apprentissage. La troisième partie concerne la phase de test qui consiste à vraiment évaluer les résultats obtenus de manière brute sans avoir d'influence sur la phase d'apprentissage.

Voici quelques exemples d'algorithmes d'apprentissage supervisé :

— K plus proches voisins (k-NN)
— Régression linéaire
— Régression logistique
— Machines à Vecteurs de Support (SVMs)
— Arbres de décision (DTs)
— Forêts aléatoires (RFs)
— Réseaux de neurones "feed-forward" (FFNNs)
— Réseaux de neurones récurrents (RNNs)
— Réseaux de neurones convolutionnels (CNNs)

1.2.1.2 L'apprentissage non-supervisé

Dans l'apprentissage non supervisé, les données ne sont pas étiquetées. Par analogie avec l'humain, le système tente d'apprendre sans professeur. Plusieurs principales familles existent dans l'apprentissage non supervisé : le partitionnement (connu sous le terme anglais *clustering*), la réduction de dimension, les règles d'association et les modèles génératifs.

Le *clustering* est une méthode permettant de repérer des similarités entre les données et de les regrouper dans une même catégorie. A l'intérieur de chaque grappe, les données sont regroupées selon une ou plusieurs caractéristiques qui leurs sont communes. Par exemple, prenons un ensemble de données représentant une population animale. Un ensemble de caractéristiques comme l'âge, le poids et la taille ont été récoltées. L'objectif sera de déterminer si certains animaux peuvent appartenir à la même espèce et ainsi les regrouper ensemble. Une des mesures de qualité d'une méthode de *clustering* sera la capacité du modèle à découvrir certains motifs cachés.

La réduction de dimension consiste à prendre des données dans un espace de grande dimension et à les remplacer dans un espace plus petit. Pour que l'opération soit utile, il faut que les données de sortie représentent les données d'entrées en terme d'information. Il est souvent recherché un optimum entre la quantité d'informations perdues et la réduction obtenue.

La recherche de règles d'association est une méthode populaire dont le but est de découvrir des relations ayant un intérêt pour le statisticien, entre deux ou plusieurs variables stockées dans de très importantes bases de données.

Les modèles génératifs sont relativement nouveaux, ils permettent la génération de données selon un motif ou catégorie apprise auparavant. Autrement dit, ils sont capables d'apprendre une distribution de probabilité sur des objets complexes, distribution que l'on pourra ensuite échantillonner pour produire des exemplaires inédits mais ressemblant aux exemples.

Voici quelques exemples d'algorithmes d'apprentissage non supervisés :

- Partitionnement
  - K-moyennes
  - Partitionnement hiérarchique
  - Maximum de vraisemblance
- Réduction de dimension
  - Analyses en Composantes Principales (ACPs)
  - Méthodes t-distributed Stochastic Neighbor Embedding (t-SNE)
  - Auto-encodeurs sans décodeur lors de la phase d'inférence
- Apprentissage de règles d'association
  - À priori
  - Éclat
- Modèles génératifs
  - Réseaux adverses génératifs (GAN)
  - Auto-encodeurs avec décodeur lors de la phase d'inférence

1.2.1.3 L'apprentissage semi-supervisé

L'apprentissage semi-supervisé se situe entre le supervisé et le non-supervisé. En effet, dans ce type d'apprentissage, un ensemble de données partiellement étiquetées est fourni. Cette méthode permet d'obtenir des modèles avec un degré de liberté plus important. En effet, des données lui sont transmises étiquetées, permettant au modèle d'extraire des caractéristiques qui pourront être réutilisées pour les données non étiquetées. Ce type d'apprentissage est souvent utilisé pour des ensembles de données conséquents. Plus la base de données est conséquente, plus il est fastidieux et chronophage d'étiqueter l'intégralité des données, cette méthode permet de contourner le problème.

La plupart des algorithmes d'apprentissage semi-supervisé sont donc des combinaisons d'algorithmes non supervisés et supervisés. Par exemple, les réseaux de conviction profonde (DBNs) s'appuient sur des composantes non supervisées appelées Machines de Boltzmann Restreintes (RBMs) et empilées les unes sur les autres. Ces RBMs sont entrainées séquentiellement en mode non supervisé, puis l'ensemble complet est optimisé précisément en utilisant des techniques d'apprentissage supervisé.

1.2.1.4 L'apprentissage par renforcement

L'apprentissage par renforcement est une approche différente. Le système d'apprentissage est un agent présent dans un environnement, il devra être capable de sélectionner, accomplir des actions et il sera amené à prendre des décisions. Cet agent va se retrouver dans une série d'états où il aura à sa disposition un ensemble d'actions. L'objectif de cet agent est de prendre la meilleure décision possible. Il doit alors apprendre par lui-même, quelle est la meilleure stratégie ou politique, pour obtenir en finalité autant de récompenses que possible. Une politique définit quelle action l'agent doit choisir face à une situation donnée.

1.2.2 Modalités d'apprentissage

Un deuxième critère utilisé pour classer les systèmes d'apprentissage automatique consiste à savoir s'ils peuvent ou non apprendre progressivement. Les modalités sont complètement indépendantes des types d'apprentissage. Autrement dit, ils ne sont pas exclusifs, il est possible de combiner les différents types avec les différentes modalités. Il existe 3 principales modalités : l'apprentissage groupé, en ligne ou par transfert.

1.2.2.1 Groupé

Dans l'apprentissage groupé, le système ne peut pas apprendre progressivement, il doit être entraîné avec l'intégralité des données disponibles. Cela nécessite en général beaucoup de temps et de ressources informatiques. Le système apprend en amont, puis une fois le modèle entrainé, il peut être utilisé en phase de production sans qu'on ne puisse faire d'apprentissage ultérieur. Autrement dit, il se contente d'appliquer ce qu'il a appris. C'est ce que l'on appelle l'apprentissage hors-ligne. Si un système d'apprentissage groupé a besoin de prendre connaissance de nouvelles données, il est

alors nécessaire d'entraîner une nouvelle version du modèle, avec l'ensemble du jeu de données tenant compte des anciennes et des nouvelles données. Une fois ce nouvel apprentissage effectué, le modèle de production sera mis à jour. L'ensemble du processus de mise à jour d'un système groupé reste relativement simple, impliquant la possibilité de le faire automatiquement. Ce type de processus permet au système de pouvoir, malgré tout, s'adapter aux évolutions.

Cette solution fonctionne très bien, mais l'entraînement sur les jeux de données complets peut prendre plusieurs heures, voire plusieurs jours, ce qui fait qu'en général, il est possible d'entraîner un nouveau système que toutes les 24h voir même seulement une fois par semaine. Toutefois, si le système a besoin de s'adapter plus rapidement ou n'est pas en capacité, en terme de ressources informatiques, de traiter toutes les données, il faudra utiliser une solution plus réactive. Pensons, par exemple, aux systèmes autonomes tels que les smartphones ou les satellites. Il ne sera ni possible de transporter un gros volume de données, ni possible de mobiliser plusieurs heures de ressources importantes pour l'entraînement.

1.2.2.2 En ligne

Une alternative intéressante aux problèmes énoncés, au paragraphe précédent, est l'apprentissage en ligne. Le système est entraîné progressivement en l'alimentant peu à peu avec des nouveaux échantillons ou observations, soit une par une, soit par petits groupes appelés mini-lots. Chaque étape d'apprentissage est rapide et économique, ce qui permet au système d'apprendre à partir de nouvelles données au fur et à mesure de leur arrivée. Les algorithmes d'apprentissage en ligne, permettent aussi d'entraîner des systèmes sur des jeux de données extrêmement volumineux ne pouvant pas tenir en mémoire. Le nom donné pour cette tâche est l'apprentissage hors-mémoire.

L'algorithme charge récursivement une nouvelle partie des données avant de l'apprendre jusqu'à ce qu'il ait analysé toutes les données. Un paramètre important des systèmes d'apprentissage en ligne est le rythme auquel ils doivent s'adapter à l'évolution des données. Ce paramètre est appelé le taux d'apprentissage, en anglais, *learning rate*. Pour un taux d'apprentissage élevé, le système s'adaptera aux nouvelles données et aura tendance à oublier rapidement les anciennes. Inversement, si le taux d'apprentissage est faible, alors le système aura une plus grande inertie. Il prendra plus en compte les anciennes données que précédemment et sera moins sensible au bruit présent dans les nouvelles données.

Une des grosses difficultés de cette modalité est que si de mauvaises données sont introduites dans le système, ses résultats se dégraderont progressivement. Pour réduire ce risque, il faut surveiller le comportement du système et interrompre rapidement l'apprentissage si une dégradation de résultats intervient. Il est aussi possible de surveiller les données d'entrées, par l'intermédiaire d'un algorithme de détection d'anomalies par exemple, et réagir en cas de données anormales.

1.2.2.3 Par transfert

Entraîner des modèles pour qu'ils aient des capacités semblables à celles de l'humain requiert énormément de ressources en terme de données et de temps. Pour optimiser un apprentissage, l'idée de l'apprentissage par transfert (*Transfer learning* en anglais) est apparue permettant de mutualiser les connaissances d'un modèle à l'autre. Il consiste à entrainer un modèle avec une très grande base de données et du matériel très performant. Dès lors, que la phase d'apprentissage a permis de pré-calculer les paramètres du modèle, il sera réutilisable par des machines bien moins performantes et bénéficieront de la connaissance acquise. Par exemple, dans le cas du traitement d'images, deux principales méthodes sortent du lot : le *fixed feature extractor* et le *fine tuning*.

La première consiste à garder les paramètres d'extraction fixés sur un réseau de neurones profond type VGGNet (présenté en 1.3) entraîné préalablement et bénéficier du modèle pour des nouvelles données. Seul l'apprentissage de la dernière couche entièrement connectée sera mise à jour en fonction des nouvelles données.

Le second consiste à utiliser la première méthode en améliorant cette fois les poids du réseau pré-entraîné. Pour cela, la méthode de backpropagation sera maintenue. En fonction de la problématique, il sera possible de garder ou figer une partie des poids des premières couches du réseau pré-entraîné car les premières couches contiennent des variables très génériques et, de ce fait, pertinentes pour de nombreuses tâches différentes. Plus on avance dans les couches du réseau, plus ces dernières sont spécifiques aux détails des classes contenues dans la base de données. Il peut donc être pertinent de mettre à jour les poids de ces dernières.

Il existe des méthodes de *transfer learning* pour d'autres types d'applications telles que le traitement du langage naturel par exemple. Cette modalité permet de gagner un temps conséquent sur l'apprentissage du modèle et peut permettre d'obtenir de meilleurs résultats. En effet, la base de données du modèle initial étant plus conséquente, elle a pu permettre la construction de caractéristiques plus pertinentes.

1.3 Historique de l'apprentissage automatique

L'histoire scientifique de l'IA recèle une rivalité très ancienne entre ces deux principales approches : l'IA symbolique et l'IA connexionniste. Les enjeux de cette rivalité intellectuelle sont encore vivaces avec, à la clé, l'évolution de l'un des domaines les plus complexes de l'IA : le raisonnement automatique. Il fait partie des points de blocage avant de créer la mythique IA générale, qui serait capable d'imiter puis de dépasser les capacités de raisonnement généralistes de l'homme.

L'IA symbolique s'appuie, notamment, sur des moteurs de règles qui permettent de créer des systèmes experts ou de faire de la programmation par contraintes. La terminologie de ce domaine intègre les notions de logiques d'ordre 0, du premier ordre (dite calcul des prédicats) et second ordre qui sont liées à la complexité des problèmes logiques au niveau d'ensemble d'objets. On peut aussi y associer le concept de logique

floue même si celui-ci présente la particularité d'être associé aussi bien à du raisonnement formel qu'à de l'apprentissage automatique. On appelle système expert, un logiciel capable de répondre à des questions, en effectuant un raisonnement à partir de faits et de règles connues. Il peut servir notamment comme outil d'aide à la décision.

Une frise chronologique avec les différents articles cités dans cet état de l'art est visible en 1.4.

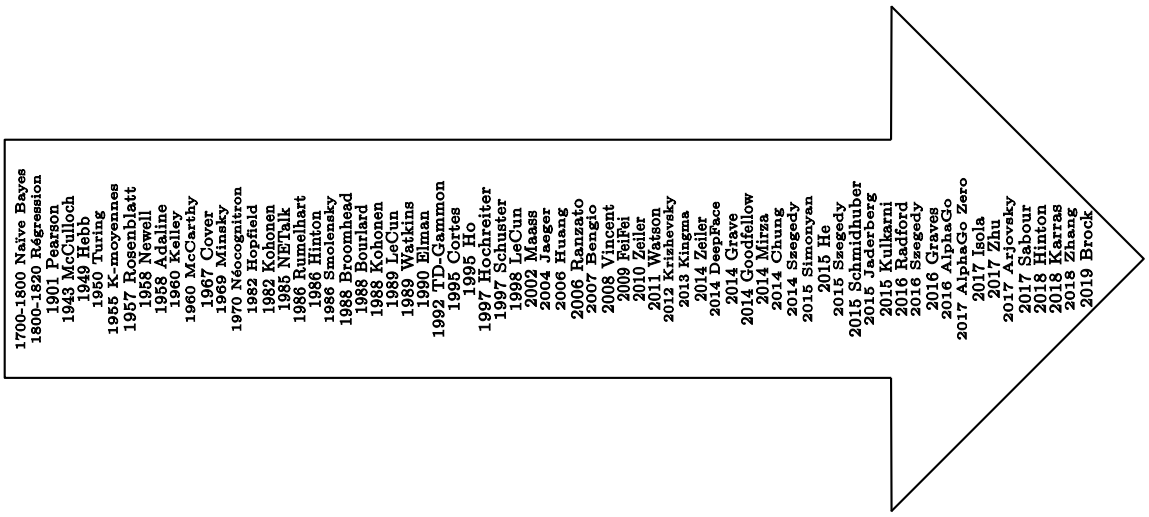


FIGURE 1.4 – Frise chronologique des principaux auteurs d'article de ce chapitre

C'est dans les années 1700-1800 que les prémices de l'IA sont lancés sans que les acteurs ne le sachent. Ce commencement peut correspondre au théorème de Bayes avec les méthodes Naïve Bayes qui en découlent. Par la suite, ce sont les méthodes de régression qui seront introduites par Legendre et Gauss début 1800.

C'est en 1901 que Pearson, publia son article précurseur sur les ACPs [19]. C'est une méthode d'analyse de données et plus généralement de statistique multivariée, qui consiste à transformer des variables liées entre elles en un nombre de variables plus faibles et décorrélées les unes des autres. Ces nouvelles variables sont appelées composantes principales.

Le premier modèle connexionniste est né en 1943 [20]. Ce premier modèle a été bio-inspiré puisque l'idée était, par l'intermédiaire de combinaisons mathématiques, de représenter ce que l'on connaissait du fonctionnement du cerveau à l'époque. Ce modèle est standard et appelé McCulloch-Pitts neurons, portant le nom de ses créateurs. Un de leurs étudiants du nom de Marvin Minsky construisit avec Dean Edmonds la première machine à réseaux neuronaux. Elle était composée de 40 neurones artificiels et son objectif était de simuler un rat cherchant sa nourriture dans un labyrinthe. Il devint l'un des plus importants leaders et innovateurs en IA que l'on pourrait qualifier de symbolique.

Quelques années plus tard, est apparue la règle de Hebb [21] ou théorie d'assemblage de neurones. Elle est à la fois utilisée comme hypothèse en neurosciences et comme concept dans les réseaux de neurones en mathématiques. Cette théorie est souvent

résumée par la formule : « des neurones qui s'excitent ensemble se lient entre eux ».

L'année d'après, le célèbre Alan Turing publia un article mémorable [22] dans lequel il énonce la possibilité de créer des machines dotées d'une véritable intelligence. Étant difficile de définir ce qu'est l'intelligence, il décida de créer ce que l'on appelle encore le test de Turing. Il consiste à dire que si une machine arrive à mener une conversation sans que l'interlocuteur ne soupçonne que ce soit une machine, alors elle sera qualifiée d'intelligente et aura passé le test de Turing.

En 1952, c'est Arthur Samuel qui créa l'un des premiers exemples de machine learning. Ses programmes ont été conçus pour jouer aux dames. Son programme était unique car, à chaque partie de dames effectuée, l'ordinateur s'améliorait toujours en corrigeant ses erreurs et en trouvant de meilleurs moyens de gagner à partir de ces données.

Trois ans plus tard, Newell et Simon conçoivent le programme Logic Theorist qui permet de démontrer automatiquement 38 des 52 théorèmes du traité Principia Mathematica d'Alfred North Whitehead et Bertrand Russel. C'est un résultat majeur et extrêmement impressionnant pour l'époque, puisque pour la première fois, une machine est capable de raisonnement. On considère légitimement ce programme comme la toute première IA de l'histoire. Quelques années plus tard, Newell et Simon vont généraliser cette approche et concevoir le General Problem Solver (GPS) [23] qui permet de résoudre n'importe quel type de problème, pour peu que l'on puisse le spécifier formellement à la machine.

C'est à cette époque que Lloyd a proposé la méthode K-moyennes. C'est la méthode de partitionnement la plus connue. Prenant en compte des points de données et un entier k, le problème est de diviser les points en k groupes, souvent appelés clusters, en minimisant une certaine fonction. On considère la distance d'un point à la moyenne des points de son cluster. La fonction à minimiser est la somme des carrés de ces distances.

Ensuite, Frank Rosenblatt a construit en 1957 [24], une idée qu'est le perceptron apparaissant sur la Figure 1.5 qu'il a plutôt matérialisé mais qui a semé les graines d'un apprentissage ascendant puisqu'il est largement reconnu comme étant le fondateur des réseaux de neurones profonds (DNNs).

L'année suivante, il publie ce que l'on appelle aujourd'hui le réseau de neurones à propagation avant ou Feed Forward Neural Network (FFNN) [25]. C'est un réseau de neurones acycliques se distinguant ainsi des RNNs. Le plus connu d'entre eux est le perceptron multi-couches qui est une extension du perceptron. On détermine les paramètres des FFNNs par rétropropagation, en donnant aux réseaux des ensembles de données appariés entre les entrées et sorties. L'erreur de prédiction appelée aussi erreur de rétropropagation dans ce cas, est souvent une variation de la différence entre l'entrée et la sortie. Étant donné que le réseau possède suffisamment de neurones cachés, il peut théoriquement toujours modéliser la relation entre l'entrée et la sortie. Pratiquement, ils sont généralement combinés avec d'autres réseaux pour en créer de nouveaux.

La même année, Widrow et Hoff développent le modèle AdaLinE. Dans sa structure,

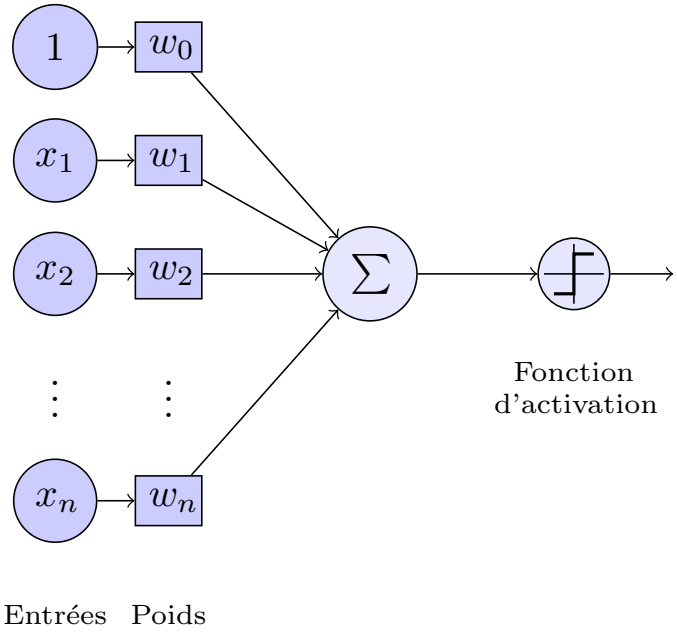


FIGURE 1.5 – Modèle du perceptron

le modèle ressemble au perceptron. Cependant, la loi d'apprentissage est différente. Celle-ci est à l'origine de l'algorithme de rétropropagation du gradient largement utilisé aujourd'hui.

En 1959, les neurophysiologistes et lauréats du prix Nobel David H. Hubel et Torsten Wiesel ont découvert deux types de cellules dans le cortex visuel primaire : les cellules simples et les cellules complexes. De nombreux Réseaux de Neurones Artificiels (RNAs) s'inspirent d'une manière ou d'une autre de ces observations biologiques. Bien qu'il ne s'agisse pas d'un jalon important pour le DL, cela a fortement influencé le domaine.

Un an plus tard. M.Kelley a écrit un article majeur et largement reconnu dans le domaine de l'aéronautique sur un algorithme d'optimisation [26]. Bon nombre de ses idées sur la théorie du contrôle, le comportement des systèmes avec entrées et la manière dont ce comportement est modifié par rétroaction ont été appliquées directement aux IAs et RNAs au fil des années. Cet article a permis le développement des algorithmes d'optimisation de rétropropagation du gradient encore largement utilisés de nos jours avec les RNAs.

La même année, McCarthy, un des deux principaux pionniers de l'IA de l'époque avec Minsky, incarnait le courant mettant l'accent sur la logique symbolique. Il a sorti son article développant les S-expressions et les S-fonctions [27] qui font appel à la récursivité. Deux années auparavant, il inventa le langage LISP permettant de faire cohabiter plusieurs processeurs.

La date importante suivante sera 1965, avec le mathématicien Alexey Ivakhnenko et ses associés, qui ont sans doute créé les premiers DNNs en appliquant ce qui n'était que théories et idées. Il a mis au point la méthode de traitement des données

de groupe et l'a appliqué aux réseaux de neurones. Pour cette raison, beaucoup le considèrent comme le père du DL moderne. Ses algorithmes d'apprentissage utilisaient des perceptrons multicouches profonds projetés vers l'avant (Deep Feed Forward Neural Network) utilisant des méthodes statistiques à chaque couche pour trouver les meilleures caractéristiques et les transmettre au système. Grâce à cette méthode, il a pu créer un DNN à 8 couches en 1971 et il a démontré avec succès le processus d'apprentissage dans un système appelé Alpha.

En 1967, Cover et Hart ont écrit l'algorithme du « plus proche voisin » [28], permettant aux ordinateurs de commencer à utiliser une reconnaissance de formes très basique. Ceci peut être utilisé pour déterminer un itinéraire de voyage par exemple. Une fois la ville de départ déterminée, l'algorithme trouvera les villes les plus proches à ne pas rater et composera ainsi un itinéraire.

Deux années plus tard, l'IA subit critiques et revers budgétaires car les chercheurs n'ont pas une vision claire des difficultés auxquelles ils sont confrontés. Leur immense optimisme a engendré une attente excessive avec des promesses non concrétisées. Dans la même période, le connexionisme a presque complètement été mis de côté durant 10 années dû à la critique dévastatrice de Minsky sur les perceptrons [29]. Ce livre constate plusieurs limites à ce que les perceptrons peuvent faire (par exemple, problème connu du "ou exclusif") et note plusieurs exagérations dans les prédictions de Rosenblatt. L'effet de ce livre est dévastateur, quasi aucune recherche dans le domaine du connexionisme ne sera faite pendant dix ans. On appelle cette période, le premier hiver de l'IA. Cependant, les fonds promis à l'IA se sont dirigés vers l'IA symbolique durant cette période.

Dans la fin des années 1970, le novateur Fukushima, est reconnu dans le domaine des réseaux neuronaux, grâce à la création du Néocognitron. C'est un RNA qui apprend à reconnaître les motifs visuels. Le Néocognitron a été utilisé pour la reconnaissance manuscrite de caractères, d'autres tâches de reconnaissance de formes ainsi que le traitement du langage naturel. Son travail qui a fortement été influencé par celui de Hubel et Wiesel, a conduit au développement des premiers CNNs, basés sur l'organisation du cortex visuel des animaux. C'est une variante des perceptrons multicouches conçue pour utiliser des quantités de données de prétraitement plus faibles.

Cependant, ce sont les travaux de Hopfield en 1982 [30] qui ont permis de sortir l'IA connexionniste de ce premier hiver. Au travers de son article court et clair, il présente une théorie du fonctionnement et des possibilités des réseaux de neurones. Il faut signaler la forme anticonformiste de son article. Alors que les auteurs jusqu'à présent s'acharnaient pour proposer une structure et une loi d'apprentissage avant d'étudier leurs propriétés, lui fixe préalablement le comportement à atteindre pour son modèle puis construit la structure et la loi d'apprentissage correspondant aux résultats escomptés. le modèle HN est aujourd'hui encore utilisé et visible sur la Figure 1.6 pour des problèmes d'optimisation. Son modèle correspond à un réseau de neurones récurrent (RNN) à temps discret dont la matrice des connexions est symétrique et nulle sur la diagonale. Sa dynamique est asynchrone puisqu'un seul neurone est mis à jour à chaque unité de temps.

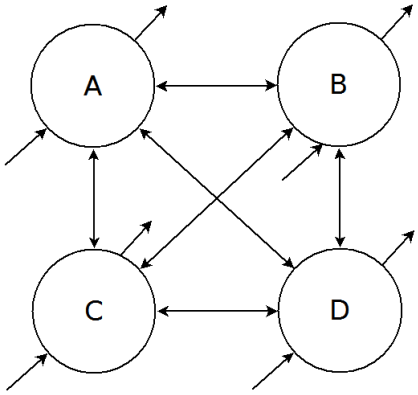


FIGURE 1.6 – Modèle du HN

Cette même année, les cartes auto-organisées ou auto-adaptatives de Kohonen (KN) [31] sont apparues. Ces réseaux utilisent l'apprentissage compétitif pour classifier les données sans supervision. L'entrée est présentée au réseau, puis il évalue lequel des neurones correspond au mieux à cette entrée. Ces neurones sont ensuite ajustés pour mieux correspondre à l'entrée, ce qui entraîne leurs voisins dans le processus.

En 1985, NETtalk a été créé par Terry Sejnowski. Son programme a été le premier à pouvoir prononcer des mots en anglais d'un même niveau environ qu'un enfant. Il a été capable de s'améliorer au cours du temps dans sa prononciation en convertissant le texte en parole.

L'année suivante, un article largement cité a été publié [32]. Il décrit plus en détail le processus de rétropropagation. Il y est montré comment améliorer considérablement l'apprentissage de réseaux de neurones connus pour de nombreuses applications telles que la reconnaissance de formes, la prédiction de mots et bien d'autres. Malgré quelques revers après ce succès, Hinton a poursuivi ces recherches au cours de ce qu'on appelle le deuxième hiver de l'IA. Il est aujourd'hui considéré par beaucoup comme étant le parrain du DL.

Cette même année, Hinton a proposé le modèle des Machines de Boltzmann (BMs) [33]. Elles ressemblent beaucoup aux HNs mais certains neurones sont marqués comme neurones d'entrées et d'autres comme neurones cachés, contrairement aux HN qui considèrent tous ses noeuds comme entrées. C'est un modèle utilisé pour l'apprentissage non supervisé. Celui-ci commence avec des poids fixés de manière aléatoire et apprend généralement avec l'algorithme de rétropropagation. Comparativement à un HN, les neurones ont pour la plupart, des fonctions d'activation binaires. Le processus d'entraînement d'un BM est assez similaire à celui d'un HN.

En suivant, Smolensky propose les RBMs [34] qui sont remarquablement similaires aux BMs et donc, également aux HN. La différence majeure entre les RBMs et les BMs est qu'elles sont plus utilisables car plus restreintes. En effet, seuls les noeuds des différents groupes sont interconnectés entre eux, un peu à l'image des FFNNs sans la couche de sortie.

C'est en 1988 que Broomhead propose les réseaux à fonction de base radiale (RBFNs) [35]. Ce sont des FFNNs avec des fonctions de base radiales telle qu'une gaussienne par exemple, comme fonction d'activation. C'est le seul changement proposé. Évidemment, il n'a pas été donné de nouveaux noms à chaque fois qu'une nouvelle fonction d'activation était proposée.

Cette même année un autre modèle à base de réseaux de neurones est proposé par Bourlard [36]. Les Auto-Encodeurs (AEs) qui sont utilisés pour l'apprentissage non supervisé. L'objectif d'un auto-encodeur est d'apprendre une représentation (encodage) d'un ensemble de données. Généralement, son but est de réduire la dimension de cet ensemble. L'ensemble du réseau ressemble toujours à un sablier horizontal comme on peut le voir sur la Figure 1.7, avec des couches cachées plus petites que les couches d'entrées et de sorties.

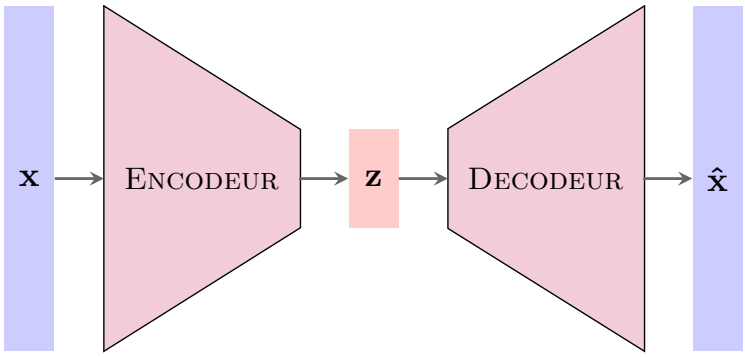


Figure 1.7 – Représentation d'un Auto-encodeur

Les AEs sont également toujours symétriques autour de la ou des couches intermédiaires (une ou deux en fonction d'un nombre pair ou impair de couches). Les couches les plus petites se trouvent presque toujours au milieu, à l'endroit où l'information est la plus comprimée correspondant aux données z dans la Figure 1.7 (point d'étranglement du réseau). La partie entre les entrées et ce point d'étranglement s'appelle l'encodage et la partie entre le point d'étranglement et les sorties s'appelle le décodage. On peut entraîner le modèle en utilisant la rétropropagation et en réglant l'erreur comme étant la différence entre les entrées et les sorties (correspondant aux entrées elles-mêmes durant la phase d'apprentissage). La partie décodage sert uniquement pour la phase d'apprentissage, cependant seule la partie encodage sera utilisée pour la phase d'inférence, réduisant ainsi la dimensionnalité des données.

De nouveau en 1988, Kohonen publia un article sur les Linear Vector Quantization (LVQ) [37]. C'est un algorithme de RNAs qui permet de choisir le nombre d'instances d'apprentissage et d'obtenir des sorties correspondant exactement aux valeurs attendues.

En 1989, c'est Yann LeCun, un autre pionnier de l'IA, qui défend la partie connexionniste de l'IA. Il a réussi à combiner les CNNs (pour lequel, il a apporté une grande contribution) avec les théories de rétropropagation, pour lire des chiffres écrits à la main [38]. Son système a finalement été utilisé pour lire les chèques manuscrits et les codes postaux par l'entreprise américaine NCR notamment, traitant entre 10 et 20 %

des chèques encaissés aux États-Unis à la fin des années 1990 et aux débuts des années 2000. Il a permis à l'architecture de CNN très connue LeNet de faire son apparition.

La même année, une avancée importante a été produite avec la thèse de Watkins [39]. Il a introduit le concept de Q-learning, qui améliore considérablement l'aspect pratique et la faisabilité de l'apprentissage par renforcement dans les machines. Cette méthode d'apprentissage a été décrite en 1.2.1.4. Un des points forts du Q-learning (largement utilisé aujourd'hui), est qu'il permet de comparer les récompenses probables et de prendre les décisions sans avoir de connaissances initiales (ou a priori) de l'environnement. Il a été prouvé, par la suite, que le Q-learning converge vers une politique optimale, c'est-à-dire, permettant de maximiser la récompense totale des étapes successives.

L'année d'après, un modèle très utilisé aujourd'hui dans la traduction des langages naturels notamment est apparu. Les RNNs présentés par Elman [40], ce sont des FFNNs avec une composante temporelle. Ils sont dotés d'états avec une notion de mémorisation qui évoluent au cours du temps. Les neurones reçoivent des informations non seulement de la couche précédente mais aussi d'eux-mêmes depuis leur valeur calculée à l'itération temporelle précédente comme montrée sur la Figure 1.8. Cela signifie que l'ordre des informations transmises en entrée et durant l'apprentissage a son importance.

La recherche des poids est complexe et peut donner lieu à des problèmes numériques tels que l'annulation ou l'explosion du gradient. Les RNNs peuvent en principe être utilisés dans de nombreuses applications car, si les bandes sonores et les vidéos sont par définition propices à ce type de modèle, dès lors que les données peuvent être transmises sous forme de série temporelle ou spatiale, alors ce modèle peut être utilisé. C'est entre autre pour cela qu'il est utilisé dans des applications de commande de systèmes dynamiques.

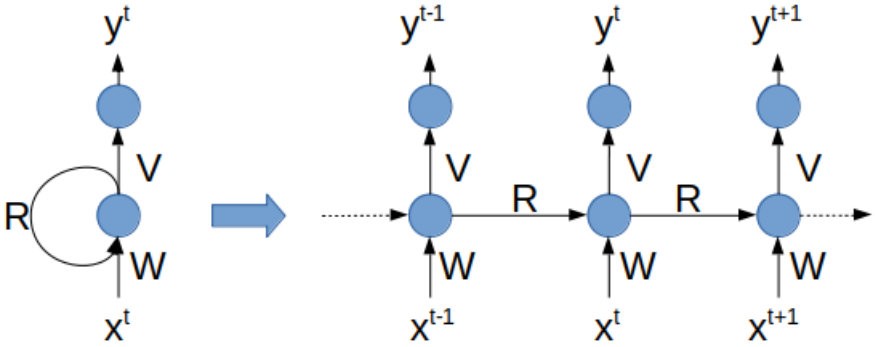


FIGURE 1.8 – Modèle du RNN

Au début des années 1990 naît l'IA en essaim, aussi appelée Intelligence Artificielle Distribuée (IAD). Cette IAD permet, par l'intermédiaire de systèmes multi-agents, de placer de manière distribuée des agents qui communiquent selon des règles établies. Chaque agent est caractérisé par le fait, qu'il est au moins partiellement autonome.

En 1992, Gerald Tesauro a proposé TD-Gammon, un programme informatique de backgammon. Il doit son nom à l'algorithme d'apprentissage par différence temporelle utilisé TD-lambda, qui a été appliqué aux RNAs. TD-Gammon a atteint un niveau

de jeu légèrement inférieur à celui des meilleurs joueurs de backgammon humains de l'époque. Il a cependant exploré des stratégies que les humains n'avaient pas poursuivis, ce qui a amené à des avancées dans la théorie du jeu de backgammon.

Les SVMs existent depuis les années 1960. Elles ont été modifiées, affinées par beaucoup au fil des décennies. Le modèle standard actuel a été conçu par Cortes et Vapnik en 1993 et présenté en 1995 [41]. Une SVM est essentiellement un système de reconnaissance et de cartographie de données similaires, et peut être utilisée pour la catégorisation de textes, la reconnaissance de caractères manuscrits et la classification d'images dans le cadre de l'apprentissage machine.

Cette même année, Ho publia un article sur les RFs [42]. Les RFs sont composées (comme le terme « forêt » l'indique) d'un ensemble de DTs. Ces arbres se distinguent les uns des autres par le sous-échantillon de données sur lequel ils sont entraînés. Ces sous-échantillons sont tirés au hasard (d'où le terme « aléatoire ») dans le jeu de données initial.

Peu après, en 1997, Hochreiter et Schmidhuber ont proposé un type de RNN avec mémoire à court et long terme (LSTM) [43]. Ils améliorent à la fois l'efficacité et l'aspect pratique des RNNs en éliminant le problème de dépendance à long terme. Autrement dit, lorsque l'information est trop lointaine dans le RNN et qu'elle est perdue. Les réseaux LSTMs peuvent mémoriser cette information sur une plus longue période en contournant le problème d'annulation ou d'explosion du gradient par l'introduction de portes et d'une cellule mémoire explicitement définie. Celles-ci sont inspirées principalement par le fonctionnement des circuits électroniques plutôt que par la biologie. Chaque neurone possède une cellule mémoire et 3 portes : entrée, sortie et oubli. La fonction de ces barrières est de protéger l'information en l'arrêtant ou en la laissant passer. De plus, cette même année, les RNNs bidirectionnels ont été introduits par Schuster [44].

De nouveau en 1997, un monde s'écroule, un roi s'agenouille. Pour la première fois, un champion du monde d'échecs est battu par une machine. Garry Kasparov n'a pas vu venir son adversaire, Deep Blue, un ordinateur conçu par la société américaine IBM. L'humiliation est totale pour celui qui avait un jour clamé que l'ordinateur ne sera jamais plus fort que l'homme. Il perdait la sixième partie de ce match historique en seulement 19 coups.

L'année suivante, LeCun contribua à un nouveau progrès dans le domaine du DL avec son article [45]. L'algorithme de SGD combiné à l'algorithme de rétropropropagation est l'approche privilégiée du DL d'aujourd'hui. LeCun, dans cet article, passe en revue diverses méthodes de reconnaissance de formes et les compare à une tâche standard de reconnaissance de chiffres manuscrits. Dans cet article, LeCun y présente son modèle LeNet-5 qui est un CNN à 7 niveaux. C'est un modèle évolué de LeNet qui obtient des résultats bien meilleurs avec des tailles d'image en entrée légèrement plus grandes. Ce modèle a majoritairement été utilisé pour reconnaître les numéros manuscrits sur les chèques numérisés d'une taille de 32×32 pixels. La capacité de traiter des images à plus haute résolution nécessite des couches et des filtres de convolution plus grands, entraînant une limitation au niveau traitement informatique.

En 2002 et 2004, Maass et Jaeger publièrent deux modèles découverts indépendamment mais qui se ressemblent. Leurs modèles sont respectivement appelés machines à état liquide (LSM) et réseau d'état d'écho (ESN) [46, 47]. Ce sont des genres particuliers de réseaux de neurones dopés (SNN) et récurrents. Ces types de réseaux sont plus proches du fonctionnement du cerveau humain et ils sont souvent utilisés pour modéliser des systèmes en rapport avec la biologie. Chaque neurone peut recevoir des signaux d'entrées dépendant du temps ou des signaux de neurones cachés. L'organisation de chaque couche neuronale est faite de manière aléatoire. La partie récurrente de ces liens génèrent des activations spatio-temporelles aux différents noeuds. L'ensemble des noeuds connectés de façon récurrente va permettre de modéliser une grande variété de fonctions non linéaires.

En 2006, les machines d'apprentissage extrême font l'objet de l'article de Huang [48]. Elles s'inspirent essentiellement des FFNNs mais avec des connexions aléatoires. Elles ressemblent aussi aux LSMs et ESNs mais ne sont ni récurrentes ni dopées. Elles n'utilisent pas non plus l'algorithme de rétropropagation du gradient. L'apprentissage commence avec des poids aléatoires et entraîne les poids lors d'une seule étape selon l'ajustement des moindres carrés. Il en résulte un apprentissage plus simple et bien plus efficace que lorsqu'il est fait avec la rétropropagation. Cependant, le modèle est moins général en terme d'application.

La même année, un nouveau modèle d'auto-encodeur est né et a été exposé par Ranzato [49]. Les auto-encodeurs épars (SAEs) sont en quelque sorte l'opposé des AEs. Au lieu d'apprendre à un réseau à représenter un ensemble de données dans un espace dimensionnel plus petit, autrement dit un nombre de noeuds sur la couche centrale, moins important, on essaye d'encoder les informations dans un espace plus grand. Son utilisation sera la bienvenue lorsque les jeux de données représentatifs de l'application sont petits. Le modèle permettra d'en obtenir un plus grand jeu de données avec une extraction possible des détails.

L'année suivante, le grand pionnier Bengio développe le Deep Belief Network (DBN) [50]. C'est le nom que l'on donne aux architectures empilées de la plupart des RBMs ou AEs. Il a été démontré que ces réseaux peuvent être entraînés pile par pile, où chaque RBM ou AE n'a qu'à apprendre à encoder le réseau précédent. Les DBNs peuvent utilisés les algorithmes d'apprentissage de rétropropagation ou de divergence contrastive [51]. Une fois entraîné vers un état plus stable grâce à un apprentissage non supervisé, le modèle peut être utilisé pour générer de nouvelles données. Si il est entraîné avec divergence contrastive, il peut même classer les données existantes parce que l'on a appris aux neurones à rechercher des caractéristiques.

Les Auto-Encodeurs Débruiteurs (DAEs), sont des AEs où le signal d'entrée est complété avec du bruit (par exemple, rendre une image plus granuleuse). Ce modèle a été introduit par Vincent en 2008 [52] qui a gardé le calcul de l'erreur original, c'est à dire la comparaison entre la sortie (entrée bruitée) et l'entrée non bruitée. Cette méthode encourage à ne pas se focaliser sur les détails, mais à se baser sur des fonctionnalités plus larges. Cela s'avère souvent judicieux car un grand nombre de données brutes sont accompagnées de bruits.

Professeure et directrice du laboratoire d'intelligence artificielle de l'université de Stanford, Fei-Fei Li a lancé ImageNet [53] en 2009. En 2017, il s'agit d'une très grande base de données gratuite de plus de 14 millions d'images étiquetées accessibles aux chercheurs, aux enseignants et aux étudiants. L'étiquetage est nécessaire pour entraîner les réseaux de neurones à l'apprentissage supervisé. Avec cette énorme base de données, qui fait encore partie des plus utilisées à ce jour, une compétition annuelle a été mise en place. Elle permet pour les chercheurs du monde entier d'évaluer leurs algorithmes de traitement d'images et de concourir pour obtenir la meilleure précision sur plusieurs tâches de vision par ordinateur.

En 2010, les réseaux déconvolutionnels (DNs) ont vu le jour [54]. Ce sont des CNNs inversés. Imaginez donner un mot en entrée et l'entraîner à produire des images du mot en sortie.

Une année s'est écoulée et c'est Watson qui fait son apparition. C'est un système de réponse aux questions développé par IBM. Il a été confronté à Ken Jennings et Brad Rutter au mythique jeu de Jeopardy. Grâce à une combinaison d'apprentissage automatique, de traitement du langage naturel et de techniques de recherche d'information, Watson a réussi à remporter le concours en trois matchs sans avoir de connexion à Internet.

Entre 2011 et 2012, c'est Alex Krizhevsky qui a remporté plusieurs concours internationaux de DL avec la création de son modèle AlexNet [55], un CNN. AlexNet est la version améliorée de LeNet-5 (construit par Yann LeCun des années auparavant). Au départ, il ne contenait que 8 couches dont 5 couches convolutionnelles et 3 couches entièrement connectées. Son succès a donné le coup d'envoi d'une renaissance des CNNs dans la communauté du DL. Son modèle a permis de surclasser tous ses concurrents précédents avec un taux d'erreur passant de 26% à 15,3%. AlexNet avait été entraîné pendant 6 jours simultanément sur deux (GPUs) Nvidia GeForce GTX 580, expliquant leur architecture divisée en deux pipelines.

En 2012, une grande expérience en apprentissage non supervisé voit le jour. Google's Artificial Brain a proposé une expérience qui peut paraître insignifiante mais l'expérience du chat a été un grand pas en avant. À l'aide d'un réseau de neurones réparti sur des milliers d'ordinateurs (environ 16000 processeurs), l'équipe a présenté au système 10 000 000 d'images non étiquetées, prises aux hasard sur Youtube, et lui a permis d'effectuer des analyses sur ces données. Une fois la séance d'apprentissage non supervisée terminée, le programme avait appris à identifier et à reconnaître les chats avec un rendement de près de 70% supérieur à celui des tentatives précédemment effectuées avec des méthodes d'apprentissage non supervisé. Évidemment, tout n'était pas parfait sur cette étude puisque le réseau n'a reconnu qu'environ 15% des objets présentés. Cela n'a pas empêché de faire un réel pas en avant pour la communauté IA.

L'année suivante, le modèle d'Auto-Encodeur Variationnel (VAE) est apparu. Son article de référence est celui de Kingma [56]. Son modèle hérite de l'architecture de l'auto-encodeur classique, mais fait des hypothèses fortes concernant la distribution des variables latentes. Il utilise l'approche variationnelle pour l'apprentissage de la repré-

sentation latente et utilise donc un algorithme d'apprentissage spécifique appelé Bayes Variationnel de Gradient Stochastique. Seule la couche centrale séparant l'encodeur du décodeur diffère avec l'apparition de deux modules : un représentant la valeur moyenne et l'autre l'écart type.

En 2013, le concours d'ImageNet (ILSVRC) a été remporté par un nouveau CNN et a fait émerger le ZFNet [57]. L'architecture de ce modèle est semblable à celui d'AlexNet, puisque seulement quelques hyperparamètres ont été changés. La diminution de la taille des filtres de convolution sur la première couche a permis de sélectionner les caractéristiques de l'image à un niveau de résolution plus fin et l'augmentation du nombre de cartes d'activation de la 3ème, 4ème et 5ème couche convolutive a augmenté le nombre de caractéristiques pouvant être détectées par le réseau.

En 2014, La plus grande plate-forme de réseaux sociaux Facebook annonçait que son système d'apprentissage DeepFace était capable d'identifier les visages figurant sur leur plate-forme avec une précision de 97,35%. Cela n'aurait pu se faire sans une telle base de données à laquelle ils ont accès. C'est une amélioration de 27% par rapport à ce que les algorithmes précédents arrivaient à obtenir. Ce résultat rivalise avec ce que l'humain est capable de faire, puisqu'il obtiendrait un score à 97,5%.

Au cours de cette année, c'est la Machine de Turing Neuronale (NTM) [58] qui a fait son apparition. Elle peut être comprise comme étant une abstraction des LSTMs et une tentative d'ouverture de la boite noire que sont les réseaux de neurones. Au lieu de coder une cellule mémoire directement dans un neurone, la mémoire est séparée. Il s'agit de combiner la permanence du stockage numérique avec la puissance expressive de réseaux de neurones. L'idée est d'avoir une banque de mémoire adressable et un réseau de neurones qui peut lire et écrire dessus. Le terme Turing vient du fait qu'elles sont complètes : la capacité de lire, d'écrire et de changer d'état en fonction de ce qu'elle parcourt signifie qu'elles peuvent représenter tout ce qu'une machine de Turing universelle peut représenter.

Dans la même année, Ian Goodfellow a publié son article sur les GANs [59]. Ces modèles permettent de s'attaquer à l'apprentissage non supervisé, ce qui est plus ou moins, l'objectif final de la communauté de l'IA. Un GAN utilise essentiellement deux réseaux concurrents comme montré sur la Figure 1.9 : le premier recueille des données et tente de créer des échantillons indiscernables (on l'appelle le générateur), tandis que le second reçoit à la fois les données réelles et les échantillons créés (on l'appelle le discriminateur), il doit déterminer si chaque point de données est authentique ou généré.

En apprenant simultanément, les réseaux entrent en concurrence et se poussent l'un et l'autre à devenir plus rapidement compétitifs. On dit que ce modèle appartient à l'apprentissage non supervisé, ce qui est un abus de langage. En effet, on le considère de la sorte car le générateur forme des sorties virtuelles durant la phase d'apprentissage. Cependant celui-ci est quand même influencé par le discriminateur qui devra définir si l'objet a été généré ou si c'est une donnée réelle. On transmet donc une information de manière indirecte du résultat attendu (cible). De plus, ce qui est très intéressant dans ce

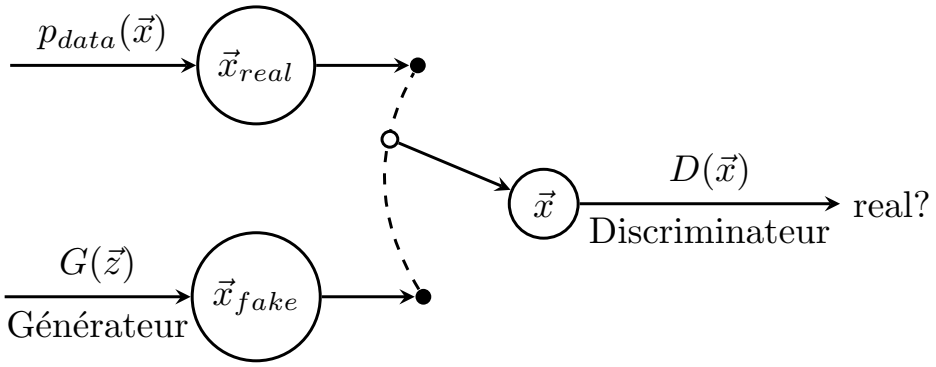


FIGURE 1.9 – Modèle du GAN

modèle, c'est que le discriminateur et le générateur peuvent être tout type d'architecture de réseaux de neurones. Ce modèle, de part son originalité et sa capacité de création, a reçu des éloges de la part de Yann LeCun en personne qui dit qu'il y a énormément de développements récents en DL, mais qu'à son avis le plus intéressant de ces 10 dernières années concerne les GANs. Ce modèle est récent puisqu'il date de 7 années, il subit déjà un essor considérable et un nombre d'articles publiés incroyables chaque année depuis sa sortie. Les articles les plus lus le concernant sont évidemment l'original référencé auparavant, mais aussi différentes architectures développées par la suite telles que le CGAN, DCGAN, Pix2Pix, CycleGAN, WGAN, StyleGAN, StackGAN, BigGAN [60-67].

Durant cette même année 2014, Chung a proposé un nouveau modèle de RNA appelé RNN à portes [68] ou Gated Recurrent Units (GRU). Ce modèle a été inspiré des LSTMs. Ils ont une porte en moins et sont câblés légèrement différemment. Au lieu de posséder une porte d'entrée, de sortie et d'oubli, ils ont une porte de mise à jour. Cette porte détermine à la fois la quantité d'informations à conserver dans son état présent et celle à laisser entrer depuis la couche précédente. La porte de remise à zéro fonctionne de la même manière que la porte d'oubli d'une LSTM, elle est juste localisée différemment. Dans la plupart des cas, leur fonctionnement est similaire à celui des LSTMs, ils permettent un fonctionnement plus efficace et plus facile à prendre en main.

Pour cette période qui fût riche en nouveautés, GoogleNet (alias Inception-V1) [69] de Google a fait son apparition en remportant le concours ImageNet en 2014. Il a atteint un taux exceptionnel d'erreur à 6,67%, très proche de la performance humaine. Le réseau est un CNN inspiré de LeNet et mis en oeuvre avec un nouveau module appelé Inception. Ce module est basé sur plusieurs petites convolutions faites en parallèle puis concaténées afin d'en retirer un grand nombre d'informations. Contrairement à ce qui se faisait précédemment, l'opération de convolution engendre des cartes d'activation de même taille que la donnée fournie afin de pouvoir concaténer les différentes maps de caractéristiques. Leur architecture consiste en un réseau CNN de 22 couches de profondeur mais le nombre de paramètres a été considérablement réduit passant de 60 millions pour AlexNet à 4 millions.

Ce modèle a, par la suite, eu plusieurs versions dont les publications [70, 71] font

références. La première concerne les modèles Inception V2 et V3 avec comme différence majeure l'amélioration du temps de calcul grâce au développement et à la factorisation des filtres de convolution. Par exemple, au lieu de mettre un filtre 5×5, on peut utiliser deux filtres 3×3 en série. De même, au lieu d'utiliser un filtre 7×7, on peut les décomposer en deux filtres 7×1 et 1×7. Cela permet un gain significatif en terme de temps de calcul. La deuxième concerne les modèles Inception-V4 et le Inception-ResNet. Ce dernier est un modèle hybride entre le modèle Inception et le ResNet exposé par la suite.

La deuxième place de ce concours a été attribuée au VGGNet [72]. Ce modèle se compose de 16 couches convolutives et il est très intéressant grâce à son architecture uniforme. Similaire à AlexNet, il fait appel à un grand nombre de filtres de convolution de taille 3×3 uniquement. C'est actuellement le choix préféré de la communauté pour l'extraction d'éléments à partir d'images. C'est un modèle composé de 138 millions de paramètres, ce qui le rend un peu difficile à gérer.

En 2015, le Residual neural Network (ResNet) [73] a introduit une nouvelle architecture avec des sauts de connexions et une forte normalisation par lots. Fondamentalement, il ajoute une identité à la solution, en transportant l'ancienne entrée et en la servant brute à une couche ultérieure (saut de connexion). Il a été démontré que ces réseaux sont très efficaces pour l'apprentissage de modèles allant jusqu'à 150 couches de profondeur, beaucoup plus que les 2 à 5 couches habituelles que l'on peut s'attendre à entraîner. Grâce à cette technique, les auteurs ont pu entraîner, durant la compétition ImageNet, un réseau de neurones de 152 couches tout en ayant une complexité moindre que VGGNet atteignant un taux d'erreur de 3,57%, ce qui est mieux que la performance humaine pour cet ensemble de données.

Au cours de cette année, l'informaticien Schmidhuber a publié une revue du DL [74]. Il a passé en revue le DL supervisé, récapitulant également l'histoire de la rétropropagation, l'apprentissage non supervisé, l'apprentissage renforcé et le calcul évolutif.

Une nouvelle avancée pour cette année concerne les réseaux d'attention (AN) [75]. Ils peuvent être considérés comme une classe à part de réseaux. Ils utilisent un mécanisme d'attention permettant de cibler l'information souhaitée par l'utilisateur. Ils sont notamment utilisés dans le cas des petites bases de données car ils permettent de focaliser l'attention à des endroits pertinents. De plus, le contexte d'attention peut être visualisé, ce qui donne un aperçu précieux des caractéristiques pertinentes et contribue à l'interprétabilité des réseaux.

Cette même année, un modèle supplémentaire apparut : il s'agit des réseaux graphiques inverses convolutionnels profonds (DCIGN) [76]. Ils ont un nom quelque peu trompeur, puisqu'ils fonctionnent comme les Auto-encodeurs avec des CNNs et DNs en tant que codeurs et décodeurs respectifs.

Puis, AlphaGo, programme informatique capable de jouer au jeu de Go et développé par l'entreprise britannique Google DeepMind devient le premier programme à battre un joueur professionnel (le français Fan Hui) sur un goban de taille classique (19×19) sans handicap. Il s'agit d'une étape symboliquement forte puisque le programme joueur

de Go est jusqu'à ce jour, un défi complexe de l'IA. En Mars 2016, il bat Lee Sedol, un des meilleurs joueurs mondiaux (9ème dan professionnel). Le 27 Mai 2017, il bat le champion du monde Ke Jie qui annonce par la même occasion sa retraite. L'agorithme d'AlphaGo [77] combine des techniques d'apprentissage automatique et de parcours de graphe, associées à de nombreux entraînements avec des humains, d'autres ordinateurs et surtout lui-même. Cet algorithme sera encore amélioré dans les versions suivantes. AlphaGo Zero [78] en octobre 2017 atteint un niveau supérieur en jouant uniquement contre son ancienne version. AlphaGo Zero en décembre 2017 surpasse largement 100-0 son ancienne version, toujours par auto-apprentissage.

En 2016, les ordinateurs neuronaux différentiables (DNCs) ont été publiés [79]. Ce sont des NTMs améliorés à mémoires évolutives. Ils ont été inspirés de la manière dont l'hippocampe humain stocke ses souvenirs. L'idée est de prendre l'architecture classique de l'ordinateur Von Neumann et de remplacer le CPU par un RNN, qui apprend à lire dans la RAM. Le DNC dispose de trois mécanismes d'attention. Ces mécanismes permettent au RNN d'interroger la similarité d'un bit d'entrée avec les entrées de la mémoire. Il permet aussi d'évaluer la relation temporelle entre deux entrées, puis de percevoir si une entrée a été récemment mise à jour. Ces mécanismes permettront de garder l'information avec une meilleure fiabilité quand il n'y a plus de mémoire disponible.

En 2017, l'un des pères du machine learning, Hinton publie un nouveau modèle qui s'intitule réseau de capsules (CapsNet) [80, 81]. Pour prendre en compte, la position relative des caractéristiques dans l'image, une approche basée sur les capsules et un algorithme d'entraînement appelé « Routage dynamique entres capsules » ont été utilisés. Les neurones ne sont donc plus reliés à un seul poids représenté par un scalaire mais à un vecteur de poids. Cela permet aux neurones de transférer plus d'informations que simplement la caractéristique détectée, comme l'endroit où elle se trouve par rapport à d'autres objets.

Le modèle ayant remporté la dernière édition du concours d'ImageNet (ILSVRC) est le SENet [82]. Cette architecture est une extension des architectures InceptionNet et Inception-ResNet qui renforce leurs performances. Le modèle a gagné le concours avec un taux d'erreur à 2,25% ce qui le propulse à la place de modèle le plus efficace. Sa particularité est l'ajout d'un réseau de neurones, appelé bloc SE, pour chacune des unités de l'architecture d'origine. Ce bloc analyse la sortie de l'unité à laquelle, il est rattaché, en se concentrant exclusivement sur la dimension de profondeur. Il va chercher quelles caractéristiques sont souvent actives ensemble. Par exemple, s'il voit apparaitre un nez et une bouche, il renforcera la carte de caractéristiques à reconnaitre des yeux même si cette dernière n'avait pas un poids important. Le bloque SE est composé de trois couches : une couche de *Pooling* à moyenne globale, une couche cachée avec la fonction d'activation ReLU et une couche de sortie avec la fonction d'activation sigmoïde.

Pour finir ce bref historique, on a pu voir les progrès spectaculaires qui ont été achevés ces 10 dernières années. Que ce soit en traitement d'images avec la vision par ordinateur notamment, l'apprentissage à base de textes avec le traitement automatique

du langage, ou alors les séries temporelles pour le traitement de la parole, toutes ont subi d'énormes avancées. Cependant, de multiples difficultés sont venues entraver les avancées visant à traiter des problématiques à partir d'autres types de données structurées telles que les graphes. Le graphe est pourtant un concept mathématiques énormément utilisé, permettant de formaliser des relations entre entités. Depuis quelques années, des solutions d'apprentissage automatique émergent pour traiter cette problématique. Plusieurs modèles de Graph Neural Networks (GNNs) ont fait leur apparition permettant d'inclure les graphes à partir de modèles existants [83-85]. Cinq familles de GNNs se dégagent, les Graph Recurrent Neural Networks (GRNNs) [86], les Graph Convolutional Neural Networks (GCNNs) [87], les Graph Auto-Encodeurs (GAEs) [88] et les Graph ATtention networks (GATs) [89] inspirés des mécanismes d'attention. Cette nouvelle branche va vraisemblablement être une perspective pour les prochaines années.

1.4 Modèles connexionnistes utilisés

Maintenant que l'historique a été passé en revue, cela donne une idée de l'engouement porté à l'IA depuis une cinquantaine d'années. La communauté scientifique n'a jamais été aussi active qu'aujourd'hui, sur cette thématique. On peut l'apercevoir au nombre d'articles publiés sur ces 5 dernières annéees. Cependant pour ces travaux, on a dû faire le choix de n'utiliser que certains modèles connexionnistes. Le choix qui a été retenu se porte sur l'utilisation des réseaux de neurones classiques et des CNNs. Ce sont des modèles d'apprentissage supervisés car l'application concernée lors des travaux nous permet d'avoir une correspondance associée à chaque donnée d'entrée. Dans cette partie, ces deux modèles sont exposés.

1.4.1 Neurone biologique

Le neurone biologique est une cellule à l'aspect inhabituel que l'on retrouve principalement dans les cortex cérébraux comme montré sur la Figure 1.10.

Il est constitué d'un corps cellulaire, qui comprend le noyau et la plupart des éléments complexes de la cellule, ainsi que de nombreux liens appelés dentrites et un très long lien appelé axone. L'axone peut être juste un peu plus long que le corps cellulaire, tout comme, il peut être des dizaines de milliers de fois plus long. Près de son extrémité, il se décompose en plusieurs ramifications appelées télodendrons qui se terminent par des structures minuscules appelées synapses et reliées aux dentrites d'autres neurones. Par l'intermédiaire de ces synapses, les neurones biologiques reçoivent des autres neurones de courtes impulsions, appelées signaux.

Lorsqu'un neurone reçoit en quelques millisecondes un nombre suffisant de signaux, il déclenche ses propres signaux. Chaque neurone biologique semble donc se comporter de façon relativement simple. Cependant ces neurones sont organisés en un vaste réseau dont son fonctionnement devient beaucoup plus complexe à l'image d'une fourmilière qui est capable de faire des constructions très complexes grâce à une accumulation de tâches simples faites par chacune des fourmis. L'architecture des réseaux de neurones

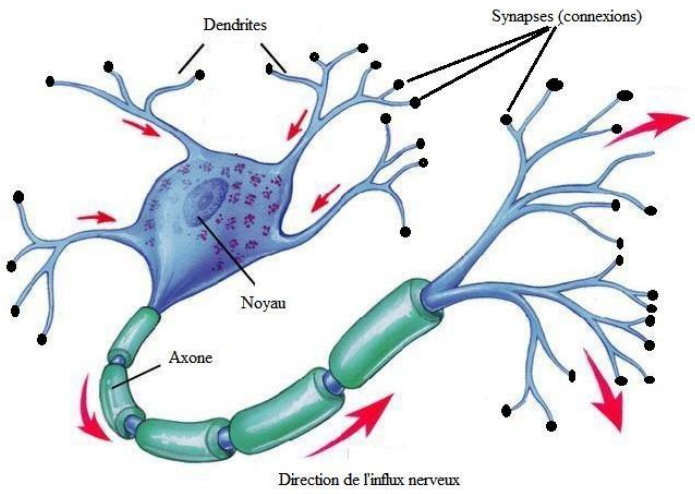


FIGURE 1.10 – Neurone biologique

biologiques fait encore l'objet d'une recherche active, certaines parties du cerveau ont été cartographiées montrant souvent une organisation sous forme de couches successives.

1.4.2 Neurone artificiel

Tout comme l'avion a été inspiré de l'oiseau, le neurone artificiel s'inspire du fonctionnement du neurone biologique. Cependant même si les avions ont pour modèle les oiseaux, ils ne battent pas des ailes. De la même manière, les RNAs sont progressivement devenus assez différents de leurs cousins biologiques.

Le fonctionnement du cerveau humain est extrêmement compliqué et est à ses prémices concernant la recherche. Même les neuro-scientifiques ne sont pas capables de déterminer tous les processus actifs dans notre cerveau et réussissent tout juste à déterminer les fonctions des différentes parties du cerveau. Pour cette raison, un modèle mathématique simplifié a été élaboré par analogie au modèle biologique comme montré sur la Figure 1.11. En effet, les différents signaux d'entrées correspondent à leurs homologues appelés dentrites, les poids associés aux entrées correspondent aux synapses, la fonction d'activation et le module de sommation des signaux sont analogues au noyau. Le signal de sortie qui peut soit être le résultat, soit être transmis à la couche de neurones suivante correspond lui à l'axone.

L'expression mathématique d'un neurone artificiel est la suivante :

$$y_{neural} = f\left(\left(\sum_{i\in\mathbb{N}} w_i * x_i\right) + b\right)$$

où w_i sont les poids associés aux entrées x_i, b un biais et f une fonction d'activation. Les poids et le biais de chaque neurone artificiel sont calculés par l'intermédiaire

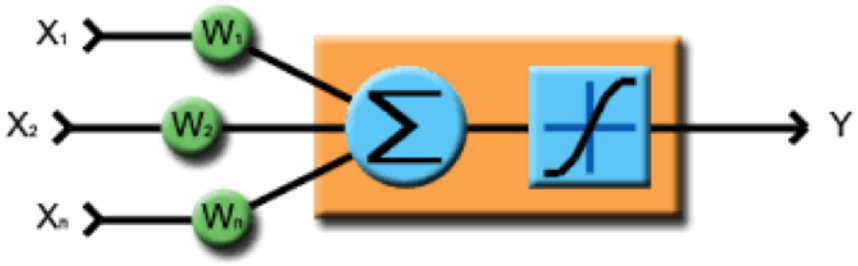


FIGURE 1.11 – Neurone artificiel

d'algorithmes d'optimisation exposés en 1.4.5. La fonction d'activation a elle aussi été inspirée du neurone biologique. Plus précisément, inspirée du potentiel d'action qui correspond à un phénomène électrique de transfert entre deux neurones biologiques. Si la force du signal électrique sortant du noyau dépasse un certain seuil, le noyau transmet le signal à l'axone. Sinon le signal est détruit par le neurone et ne se propage pas davantage. Autrement dit, le potentiel d'action correspond à la variation de la force du signal indiquant si la communication doit être établie ou détruite.

L'activation a pour objectif de transformer le signal afin d'obtenir une valeur de sortie ayant subi une combinaison complexe avec les entrées. C'est pour cela qu'elle est souvent choisie pour être une fonction mathématique non linéaire. Dans de rares cas, la fonction linéaire sera utilisée comme fonction d'activation. Son utilité est faible car si l'architecture du réseau possède plus d'une couche de neurones avec cette fonction d'activation, alors l'application récursive de cette dernière n'aura plus d'impact sur le résultat. Cependant, l'utilisation de fonctions non linéaires permet d'utiliser n'importe quelle architecture et de résoudre des problèmes complexes. De plus, dans le cas où les données sont situées dans un ensemble non linéairement séparable, l'utilisation de fonction d'activation non linéaire sera obligatoire pour obtenir un apprentissage correct.

Les fonctions d'activation les plus utilisées sont la sigmoïde, la tangente hyperbolique, la ReLU, la ReLU paramétrique, la ELU et la fonction Softmax. Chacune d'entre elles a des inconvénients et beaucoup de recherches contribuent pour trouver des alternatives. Il en existe beaucoup plus que celles citées et chacune ont des avantages et des inconvénients.

La sigmoïde de la forme $\sigma(x) = \frac{1}{1+e^{-x}}$ est historiquement celle qui était la plus utilisée. Les valeurs de sortie sont situées dans l'intervalle $I = [0; 1]$ avec une saturation pour les nombres à valeur absolue élevés en entrée. Un des inconvénients de cette fonction est qu'elle n'est pas centrée sur zéro, c'est à dire que des valeurs négatives en entrée peuvent prendre une valeur positive en sortie. De plus, elle est coûteuse en terme de calcul car elle comprend la fonction exponentielle.

La tangente hyperbolique de la forme $tanh(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$ est proche de la fonction sigmoïde mise à part que ses valeurs de sortie sont situées dans l'intervalle $I = [-1; 1]$ et que la fonction est centrée sur zéro. La ReLU de la forme $f(x) = max(0, x)$ est de nos jours la plus utilisée. Si l'entrée est négative, la sortie est à 0 et si elle est positive,

la sortie vaut x. Cette fonction augmente considérablement la convergence du réseau et ne sature pas. Un de ses inconvénients est que si la valeur d'entrée est négative, le neurone reste inactif, ainsi les poids ne sont pas mis à jour et le réseau n'apprend pas.

La ReLU paramétrique de la forme : si $x < 0$, $f(x) = ax$ avec un hyperparamètre $a \in \mathbb{R}$ et si $x \geq 0$, $f(x) = x$ est la forme générale de la fonction ReLU. Cette fonction élimine en partie le problème d'inactivité pour les valeurs négatives, cependant, les résultats obtenus ne sont pas toujours cohérents. La ELU de la forme : si $x < 0$, $f(x) = a(e^x - 1)$ avec un hyperparamètre $a \in \mathbb{R}$ et si $x \geq 0$, $f(x) = x$. Pour finir, la fonction Softmax de la forme $f(x)_j = \frac{ex_j}{\sum_{k=1}^{K} ex_k}$ avec $j \in \{1, ..., K\}$ est utilisable uniquement sur la couche de sortie et pour les réseaux de neurones utilisés en mode classification. Elle permet d'avoir un résultat sous forme de probabilité de chance d'obtenir la classe concernée.

1.4.3 Réseaux de neurones artificiels

Un RNA est un ensemble de neurones artificiels interconnectés et organisés sous forme de différentes couches comme on peut le voir sur la Figure 1.12. Les noeuds dont le préfixe est respectivement, E ,C et S correspondent à la couche d'entrée, à la couche cachée et à la couche de sortie.

La couche d'entrée permet de distribuer les données d'entrées aux neurones de la couche suivante. Ensuite, il y a une ou plusieurs couches cachées et une couche de sortie. Chaque couche à l'exception de la couche de sortie, comprend plusieurs neurones dont un ayant pour valeur un terme constant appelé biais et est intégralement reliée à la suivante. Lorsqu'un réseau de neurones contient deux couches cachées ou plus, on dira que c'est un DNN.

Le nombre de neurones des couches d'entrée et celui de sortie sont respectivement fixés par le nombre de données fournies en entrée et en sortie. Le nombre de neurones dans la ou les couches cachées est défini par l'utilisateur en fonction de l'application. Il n'y a pas de valeurs théoriques les concernant [90], ils doivent être trouvés de manière expérimentale. Cependant, il est nécessaire de correctement agréger ces hyperparamètres car ils ont un lien direct avec la généralisation du modèle. Une mauvaise calibration de ces hyperparamètres peut entrainer du sur-apprentissage.

On a vu précédemment que la réponse de chaque neurone des couches cachées et de sorties était formalisée par une expression mathématique avec des poids ou paramètres permettant de construire le modèle. Les valeurs de chaque poids sont déterminées par l'intermédiaire d'algorithmes d'optimisation qui sont présentés par la suite, en 1.4.5. Ces algorithmes vont minimiser, itérativement, l'erreur entre la valeur cible et la valeur prédite par le réseau de neurones sur un maximum d'échantillons de données durant la phase d'apprentissage. Ensuite, l'utilisation du modèle pour prédire des résultats à partir de nouvelles données d'entrées durant la phase d'inférence sera possible.

L'erreur est définie par l'intermédiaire de la fonction coût aussi appelée fonction perte, à minimiser. Cette fonction peut prendre différentes formes selon le mode

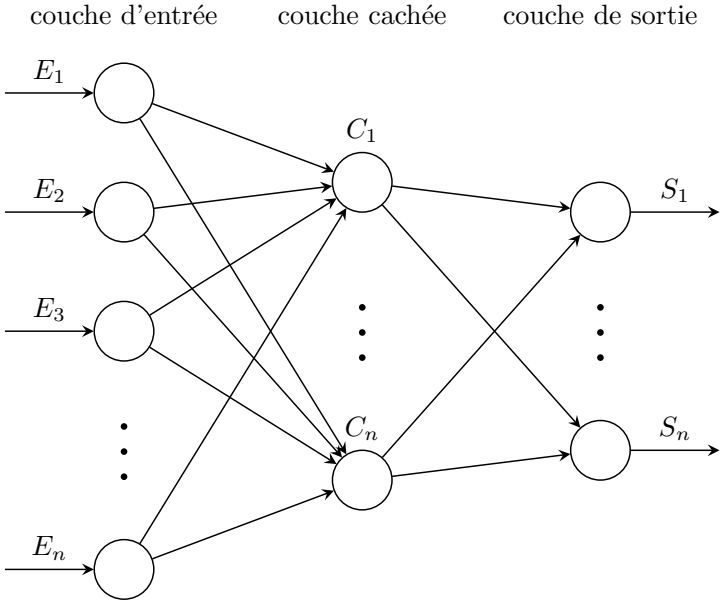


FIGURE 1.12 – Architecture du réseau de neurones feed-forward

d'apprentissage choisi. Par exemple, les fonctions de perte les plus utilisées pour le mode régression sont la fonction d'erreur moyenne au carré (MSE), la fonction d'erreur quadratique moyenne (RMSE) ou la fonction d'Erreur Moyenne Absolue (MAE). La fonction MSE à l'allure suivante :

$$\mathcal{L}(o, y) = \frac{1}{N} \sum_{i=1}^{N} (o_i - y_i)^2$$

où o correspond au vecteur de valeurs cibles, y correspond au vecteur des prédictions faites par le réseau de neurones et N au nombre d'échantillons de données. Cette fonction a été choisie pour ses propriétés quadratiques. En effet, par définition les fonctions quadratiques sont convexes. Cette catégorie de fonction est très recherchée en optimisation car elle permet de trouver des minimums globaux pour un problème donné.

Le deuxième exemple concerne les fonctions pertes les plus utilisées pour le mode classification qui sont la fonction d'entropie croisée (*cross entropy* en anglais) ou la fonction de Hinge. La fonction *cross entropy* à l'allure suivante :

$$\mathcal{L}(o, y) = -\sum_{i=1}^{N} o_i * log(y_i)$$

Les fonctions coûts énoncées précédemment ne sont pas exhaustives. Ces fonctions ont une grande importance dans le processus d'apprentissage puisque c'est d'elles que vont dépendre les valeurs des poids fixées. Cependant, ce n'est pas parce que la fonction coût utilisée est convexe que la modélisation par réseaux de neurones est convexe. En effet, chaque nœud utilise des fonctions d'activation non linéaires retirant les propriétés

de convexité au modèle. La difficulté concernant les problèmes d'optimisation non convexes est qu'il est possible d'obtenir des minimums locaux. Dans ce cas, il est nécessaire de trouver un minimum local avec une faible valeur de fonction coût afin d'être proche du minimum global [91].

Certains travaux ont montré qu'en prenant un modèle réseau de neurones simplifié, quasi tous les minima locaux étaient proches des minima globaux [92, 93]. Cependant, Choromanska et al. a montré que trouver un minimum local avec une faible valeur d'erreur n'implique pas forcément un bon apprentissage. En effet, un faible minimum local peut entraîner du sur-apprentissage et fournir un modèle non représentatif du système. Une étude [94] a tenté de montrer que les minima locaux qui généralisent ont des propriétés différentes de ceux qui sur-apprennent. Plus précisément que les bassins d'attraction des minimums locaux qui généralisent seraient plus larges que ceux qui sur-apprennent. Ceci pourrait être un facteur explicatif de la convergence de la SGD.

La recherche est active concernant les algorithmes d'optimisation dans l'objectif de rendre plus efficace le calcul de la valeur des poids du réseau durant la phase d'apprentissage. Jusqu'à présent, cette phase prend du temps et des ressources numériques. Cependant, dès lors que le modèle est fixé, la phase d'inférence est très efficace car elle fait appel à une itération de calcul neuronal propagée vers l'avant.

1.4.4 Réseaux de neurones convolutionnels

L'essor considérable apporté à l'IA est en grande partie associé à l'arrivée du DL. Huit fois sur dix, lorsque que le mot Deep Learning apparait, il fait référence aux CNNs.

Les CNNs, qui sont spécifiquement conçus pour traiter la variabilité des formes deux dimensions (2D), sont plus performants que toutes les autres techniques. Ils sont très différents des autres réseaux de neurones existants. Ils sont principalement utilisés pour le traitement de l'image mais peuvent également être utilisés pour d'autres types de données tels que l'audio par exemple. Dans la majeure partie des cas dans la littérature, ils sont utilisés pour la classification. Cependant, ils peuvent être employés pour des problèmes de régression.

Un cas typique des CNNs est celui où on transmet des images au réseau et celui-ci est capable d'identifier l'image correspondante. Les CNNs ont été développés pour faire face au problème de dimensionnalité des données. En effet, lorsque l'on souhaite classer des images par exemple, si l'image a une haute résolution, chaque pixel est considéré comme une variable d'entrée apportant une dimension considérable au modèle. La particularité des CNNs se situe dans sa manière de scanner les entrées. Contrairement à ce qui se faisait précédemment, l'intégralité des données ne seront pas traitées en même temps. En effet, un nombre de cartes ou un ensemble de caractéristiques (*feature maps* en anglais) par couche va être déterminé par l'utilisateur puis calculé par l'algorithme d'apprentissage via des filtres de convolution d'une taille donnée. Cela permettra de définir des caractéristiques bien précises des images permettant de classifier les différents objets et de les reconnaître.

Un des avantages d'agir de la sorte est que peu importe la localisation de l'objet sur l'image ; s'il possède des caractéristiques décodées par l'algorithme alors il sera détecté sur de nouvelles images. Cependant, un des inconvénients de ce type de modèles est qu'il n'y a pas de lien relatif à la position inter-objet. Autrement dit, une bouche et un nez pourront être détectés sur un visage, cependant, l'algorithme ne saura pas que le nez se situe toujours au-dessus de la bouche.

L'architecture classique des CNNs est exposée en Figure 1.13. Plusieurs couches peuvent faire partie de l'architecture, les couches de convolution, les couches de *Pooling*, les couches ReLU correspondant aux fonctions d'activations vues en 1.4.2. Toutes ces couches sont empilables plusieurs fois, décrivant la profondeur du réseau. Pour terminer, une couche entièrement connectée (FC) avec une fonction d'activation Softmax pour la couche de sortie, à l'image d'un FFNN qui est empilée à la suite du réseau. Elle permet de rassembler, par l'intermédiaire d'une concaténation, toute l'information provenant des cartes de caractéristiques de la dernière couche et fournir un résultat en sortie.

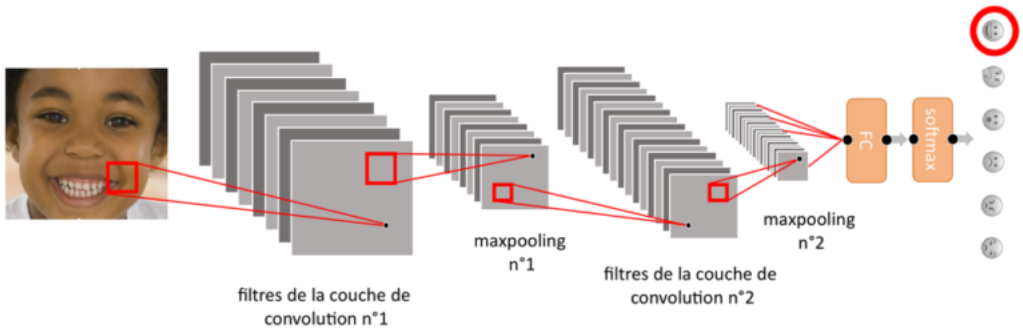


FIGURE 1.13 – Architecture classique d'un CNN

Les *feature maps* des différentes couches sont obtenues par l'intermédiaire d'un produit de convolution entre les *feature maps* des couches précédentes ou les données d'entrées (si la 1ère couche est considérée), et un filtre de convolution. Elles sont calculées durant la phase d'apprentissage et permettent de dégager les caractéristiques des ensembles de données. Le calcul de ces *feature maps* est montré avec un exemple sur la Figure 1.14. La première étape est de prendre le filtre de convolution et de le placer en haut à gauche de l'image à filtrer. Une fois la valeur du premier pixel de la carte de caractéristiques obtenue, le filtre de convolution viendra glisser sur les pixels suivants.

Un paramètre appelé *stride* permet de déterminer le pas de glissement entre chaque pixel considéré. Par exemple, si le *stride* est à 1 alors tous les pixels sont considérés. S'il est défini à 2, un pixel sur deux est considéré, réduisant la taille de la carte de caractéristiques ainsi obtenue.

Un deuxième paramètre concerne le *padding* qui permet de garder la même taille pour la carte de caractéristique que l'image à filtrer lorsqu'un *stride* de 1 est utilisé. En effet, le produit de convolution sans *padding* réduira la taille de l'image selon la taille du filtre utilisé. Si un filtre 3×3 est pris en compte, le premier pixel de l'image à filtrer considéré est celui de la $2^{\text{ème}}$ ligne et $2^{\text{ème}}$ colonne entrainant une réduction de la carte

de caractéristiques d'une dimension (2×2). Lorsque le *padding* est sélectionné, alors une marge de 0 est placée sur les dimensions manquantes.

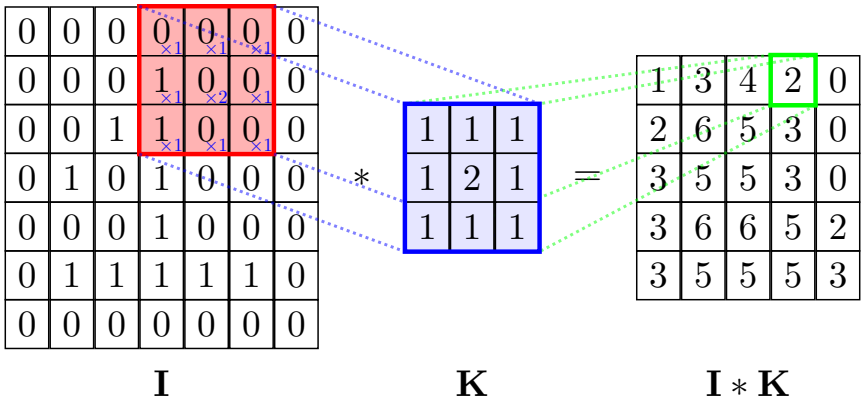


FIGURE 1.14 – Représentation du produit de convolution entre deux matrices

De plus, il existe d'autres possibilités de calcul des cartes de caractéristiques en appliquant les filtres de différentes manières, on peut par exemple décider d'appliquer une convolution éclatée, c'est à dire, qu'on considère un pixel sur deux pour calculer la convolution. Autrement dit, pour un filtre 3×3, on considère 5×5 pixels de l'image à traiter avec des pixels à l'intérieur qui ne seront pas considérés. Un taux de dilatation permet de régler quel est le nombre de pixels qui ne seront pas pris en compte.

Ces filtres à convolution sont mis à jour itérativement durant la phase d'apprentissage. Au niveau de la phase d'apprentissage, l'optimisation se fait de la même manière que présentée en 1.4.5. En effet, on peut considérer chaque valeur de chaque filtre comme étant des poids reliant les neurones d'une couche précédente vers une couche suivante. Considérons que sur la Figure 1.13, les images en entrée sont de dimension 28×28, la première couche de convolution possède des filtres de taille 5×5 et un nombre de cartes de caractéristiques de 8, la deuxième couche possède des filtres de taille 3×3 et un nombre de cartes de caractéristiques de 16, la couche entièrement connectée possède 64 neurones sur la couche cachée et 6 sur la couche de sortie correspondant aux différentes classes. Le *stride* est de 1 et il n'y a pas de *padding*. Alors le nombre de poids du réseau à calculer est de 27430. Le détail du calcul de cette valeur est donné dans le Tableau 1.1

Une couche de *Pooling* peut aussi être appliquée. Le *Pooling* est une opération simple qui consiste à remplacer un carré de pixels (généralement 2×2 ou 3×3) par une valeur unique. Cette opération permet de diminuer considérablement la taille des fenêtres de caractéristiques et d'en garder l'information importante. Tout comme pour la couche de convolution, la fenêtre de *Pooling* va se déplacer sur toute l'image avec la possibilité de lui appliquer un pas de glissement (*stride*). Il existe plusieurs types de *Pooling* dont les principaux sont :

— Le max-*Pooling*, qui revient à prendre la valeur maximale de la sélection comme montré en Figure 1.15. C'est le type le plus utilisé car il est rapide à calculer et permet de simplifier efficacement l'image. Il a tendance à retenir les caractéristiques

	couche	taille	nombre de paramètres
0	entrées	28×28	0
1	conv 1 (5×5)	28-(5-1)=24 $\rightarrow$ $24 \times 24 \times 8$	(5*5+1)*8 = 208
2	max *Pooling* 1 (2×2)	$12 \times 12 \times 8$	0
3	conv 2 (3×3)	12-(3-1)=10 $\rightarrow$ $10 \times 10 \times 16$	(8*(3*3)+1)*16 = 1168
4	max *Pooling* 2 (2×2)	$5 \times 5 \times 16$	0
5	dense	64	(16*5*5+1)*64 = 25664
6	sortie	6	(64+1)*6 = 390

Tableau 1.1 – Nombre de poids intervenant sur le réseau

les plus marquées de la sélection de pixels.

— Le mean *Pooling*, qui correspond à la valeur moyenne des pixels de la sélection. La somme de toutes les valeurs divisée par son nombre est effectuée. Il a tendance à faire sortir les caractéristiques moins marquées.

— le sum *Pooling*, il ressemble beaucoup au mean *Pooling*. Seule la somme des valeurs de la sélection est effectuée.

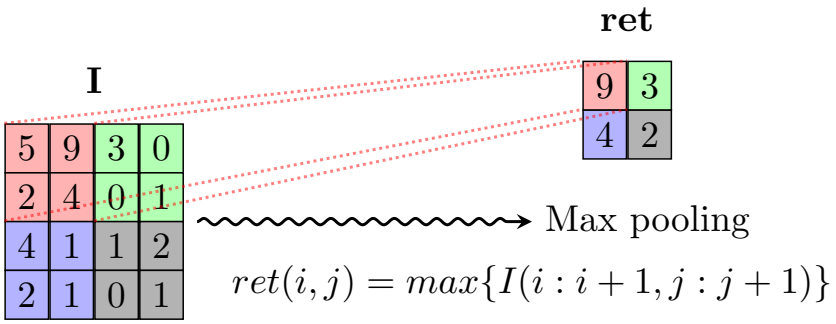


FIGURE 1.15 – Représentation de l'opération max-*Pooling*

Des couches dites de ReLU sont également applicables. Ce sont en réalité des couches d'activation dont le terme de ReLU en est la fonction. Cette fonction est appliquée terme à terme à la matrice composant l'image. Ce sont les mêmes fonctions que vues précédemment en 1.4.2. Des couches de régularisation comme le dropout exposé en 1.2.1.1 peuvent aussi être impliquées dans ce type de modèle.

Pour finir, la couche de mise à plat (*flattening* en anglais) a son importance. Elle va concaténer tous les pixels formant les cartes de caractéristiques de la dernière couche du modèle. Elle va permettre de transmettre les informations extraites par cette première partie du modèle au réseau entièrement connecté. Le fonctionnement du réseau entièrement connecté a été décrit en 1.4.3. La sortie de ce type de modèle est dépendante de la fonction d'activation attribuée à la dernière couche. Dans la plupart des applications, correspondant à la classification de données, la fonction d'activation est le Softmax.

Le modèle exposé dans cette partie est classique. Les différentes couches présentées ne sont pas exhaustives et peuvent être empilées constituant l'architecture du réseau. Architecture dont on a vu en 1.3 qu'il en existait un certain nombre dans la littérature.

Des contraintes majeures dans les modèles connexionnistes correspondent aux hyperparamètres. Ils sont un grand nombre à devoir être fixés manuellement et expérimentalement en fonction du problème posé. Par exemple, pour les CNNs, les hyperparamètres suivants sont à déterminer :

— Nombre de couches à empiler
— Nombre de cartes de caractéristiques
— Taille de chaque filtre de convolution
— Taille de la fenêtre de *Pooling*
— Le pas de glissement de la fenêtre
— Le nombre de neurones cachés dans la couche entièrement connectée
— Le taux d'apprentissage
— Ratio de distribution des données d'apprentissage (entrainement, validation et test)

Ceci n'est pas une énumération exhaustive de tous les hyperparamètres à déterminer mais on peut imaginer la portée de la complexité du modèle. Pour cette raison, des modèles ont été établis comme cités en 1.3, permettant de décrire quelques idées de conduite à tenir pour déterminer ces hyperparamètres.

1.4.5 Algorithmes d'optimisation

L'optimisation est un vieux domaine des mathématiques consistant à trouver la meilleure solution à un problème. Généralement, il n'est pas possible de tester toutes les solutions possibles et de voir celle qui fonctionne le mieux pour des raisons de temps de calcul ou de budget. Pour palier cela, on modélise le problème sous forme mathématique, afin d'obtenir la meilleure solution. Dès lors qu'elle a été trouvée, il faut vérifier que le modèle est représentatif du système. Évidemment, plusieurs modélisations et différentes techniques d'optimisation peuvent être utilisées pour un même problème. Il s'agit d'utiliser la plus cohérente ou appropriée en fonction du problème posé.

Lorsque nous utilisons un réseau de neurones à propagation avant, l'entrée x fournit l'information initiale, qui se propage ensuite à travers les unités cachées de chaque couche et fournit en sortie une prédiction. On appelle cette étape la propagation vers l'avant. Durant la phase d'apprentissage, le résultat prédit va être comparé avec le résultat attendu (cible) et sera évalué via une fonction coût. L'algorithme de rétropropagation permet par l'intermédiaire de la fonction coût de propager l'erreur en arrière dans le réseau et via le calcul d'optimisation choisi, de mettre à jour les paramètres du réseau afin de faire décroître cette fonction coût. Le terme de rétropropagation est souvent interprété à tort comme qualifiant l'ensemble de l'algorithme d'apprentissage pour des

réseaux de neurones multicouches. En réalité, la rétropropagation concerne uniquement la méthode de propagation de l'erreur en arrière tandis qu'un autre algorithme, tel que la SGD, est utilisé comme une surcouche effectuant l'apprentissage en modifiant de façon itérative les paramètres (poids et biais) des connexions du réseau.

1.4.6 Algorithme de rétropropagation de l'erreur

Pour commencer, il est important de définir les différents symboles à l'aide de la Figure 1.16, qui seront utilisés par la suite :

- $L \in \mathbb{N} \setminus \{0\}$ est le nombre de couches du réseau de neurones
- σ est la fonction d'activation
- w^l est une matrice de taille $n \times m$ qui contient tous les poids de la l$^{\text{ème}}$ couche avec $n,\ m \in \mathbb{N}$ et $l \in [\![1, L]\!]$
- w^l_{ij} est le poids qui relie le i$^{\text{ème}}$ neurone de la l$^{\text{ème}}$ couche au j$^{\text{ème}}$ neurone de la (l-1)$^{\text{ème}}$ couche
- b^l est le vecteur qui contient tous les biais de la l$^{\text{ème}}$ couche
- b^l_i est le biais (entrée de valeur constante) associé au i$^{\text{ème}}$ neurone de la l$^{\text{ème}}$ couche
- z^l_i est la valeur d'agrégation du i$^{\text{ème}}$ neurone de la l$^{\text{ème}}$ couche, c'est à dire la valeur qu'un neurone obtient avant son passage à la fonction d'activation
- z^l est le vecteur qui contient toutes les agrégations de la l$^{\text{ème}}$ couche
- a^l_i est la valeur d'activation du i$^{\text{ème}}$ neurone de la l$^{\text{ème}}$ couche, c'est à dire le résultat de la fonction d'activation appliquée à la valeur de l'agrégation z^l_i
- a^l est le vecteur qui contient toutes les activations de la l$^{\text{ème}}$ couche
- $C(w, b)$ correspond à la fonction coût qui est tributaire des poids et des biais

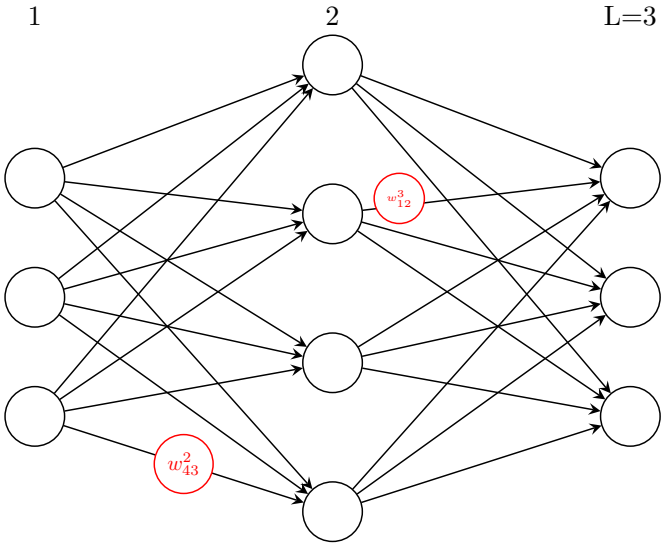


FIGURE 1.16 – Structure d'un réseau de neurones

Maintenant que les différents symboles ont été introduits, les équations mathématiques de la rétropropagation de l'erreur vont être décrites. Les différentes variables ou

paramètres construisant le modèle des réseaux de neurones sont les différents poids et biais. Ce sont eux qui vont être actualisés avec des méthodes d'optimisation afin de créer un système en cohérence avec les données fournies, cohérence qui est représentée comme le minimum de la fonction coût $C(w, b)$.

La rétropropagation de l'erreur implique l'utilisation de dérivées partielles afin de répartir de manière optimisée l'erreur en tout point du réseau. Le but étant de minimiser la fonction coût en jouant sur les variables du réseau, il va falloir calculer les expressions $\frac{\partial C}{\partial w_{ij}^l}$ et $\frac{\partial C}{\partial b_i^l}$.

Habituellement, si la fonction est simple, il suffit d'appliquer les règles de dérivation en s'aidant de quelques dérivées usuelles pour obtenir une dérivée en bonne et due forme. Cependant, un réseau de neurones possède une représentation constituée, potentiellement, de milliers de fonctions composées entre elles dont la dérivée n'est pas simple à calculer. Heureusement, il existe la solution appelée rétropropagation.

Pour les lecteurs non familiers avec le calcul des dérivées partielles, on rappelle dans ce qui suit la définition. La dérivée partielle d'une fonction de plusieurs variables est sa dérivée par rapport à l'une de ses variables, les autres étant gardées constantes. Prenons un exemple concret et on va essayer de calculer $\frac{\partial C}{\partial w_{11}^L}$ avec w_{11}^L qui correspond au lien entre le 1er neurone de l'avant dernière couche et le 1er neurone de la dernière couche. Par exemple, modifions la valeur de w_{11}^L en y ajoutant une petite valeur qu'on appellera Δw_{11}^L. En modifiant ce poids, on modifie la valeur de sortie du réseau et donc la valeur calculée par la fonction coût. Cette modification de la valeur de $C(w, b)$, on la notera ΔC. Par définition, lorsque l'on fait tendre cette valeur vers 0, on obtient l'équation :

$$\frac{\partial C}{\partial w_{11}^L} \Delta w_{11}^L = \Delta C$$

Car on rappelle que $\frac{\partial C}{\partial w_{11}^L}$ mesure la variation du coût en fonction de w_{11}^L.

On remarque qu'il n'y a pas de lien direct entre la fonction coût et le poids concerné. C'est pour cette raison que l'on va être contraint d'utiliser la règle de dérivation en chaîne pour obtenir l'expression :

$$\frac{\partial C}{\partial w_{11}^L} = \frac{\partial z_1^L}{\partial w_{11}^L} \frac{\partial a_1^L}{\partial z_1^L} \frac{\partial C}{\partial a_1^L}$$

Pour faire simple, afin de calculer comment w fait varier la fonction C, il suffit de calculer comment w fait varier z, comment z fait varier a et comment a fait varier C. Maintenant, étudions les termes séparément.

$\frac{\partial C}{\partial a_1^L}$ est la variation du coût en fonction de la sortie du réseau. Cela correspond à la dérivée de la fonction coût par rapport à la seule variable a_1^L, les autres étant constantes par ailleurs :

$$\frac{\partial C}{\partial a_1^L} = C'(a_1^L)$$

$\frac{\partial a_1^L}{\partial z_1^L}$ désigne la variation de la fonction d'activation dépendant de l'agrégation. Pour obtenir cette valeur, il suffit de calculer la dérivée de la fonction d'activation :

$$\frac{\partial a_1^L}{\partial z_1^L} = \sigma'(z_1^L)$$

Pour finir, $\frac{\partial z_1^L}{\partial w_{11}^L}$ est la variation de la fonction d'agrégation dépendant de ce seul poids. En utilisant les règles usuelles de dérivation, on obtient :

$$\frac{\partial z_1^L}{\partial w_{11}^L} = a_1^{L-1}$$

On obtient donc l'équation :

$$\frac{\partial C}{\partial w_{11}^L} = a_1^{L-1} * \sigma'(z_1^L) * C'(a_1^L)$$

que l'on peut généraliser :

$$\frac{\partial C}{\partial w_{ij}^L} = a_j^{L-1} * \sigma'(z_i^L) * C'(a_i^L)$$

Dans un but de simplification pour la suite, on pose :

$$\delta_i^L = \sigma'(z_i^L) * C'(a_i^L) = \frac{\partial a_i^L}{\partial z_i^L}\frac{\partial C}{\partial a_i^L} = \frac{\partial C}{\partial z_i^L}$$

Et on a :

$$\frac{\partial C}{\partial w_{ij}^L} = a_j^{L-1} * \delta_i^L$$

Considérons maintenant le poids particulier w_{12}^{L-1} de la couche précédente $L-1$:

$$\frac{\partial C}{\partial w_{12}^{L-1}} = \frac{\partial z_1^{L-1}}{\partial w_{12}^{L-1}}\frac{\partial a_1^{L-1}}{\partial z_1^{L-1}}\frac{\partial C}{\partial a_1^{L-1}}$$

Le premier terme $\frac{\partial z_1^{L-1}}{\partial w_{12}^{L-1}}$ est la variation d'agrégation dépendant de ce poids. En suivant le même raisonnement que précédemment, on a a_2^{L-2}.

Pour le deuxième terme $\frac{\partial a_1^{L-1}}{\partial z_1^{L-1}}$, on a $\sigma'(z_1^{L-1})$.

Enfin il reste le dernier terme $\frac{\partial C}{\partial a_1^{L-1}}$. Pour le calculer, utilisons de nouveau la chaîne de dérivation. Il est nécessaire de remarquer que si a_1^{L-1} est modifié, cela va avoir un impact sur les valeurs de $\{z_1^L, ..., z_n^L\}$ avec n le nombre de neurones de la couche L comme décrit précédemment. On obtient l'équation suivante :

$$\frac{\partial C}{\partial a_1^{L-1}} = \frac{\partial z_1^L}{\partial a_1^{L-1}} \frac{\partial C}{\partial z_1^L} + ... + \frac{\partial z_n^L}{\partial a_1^{L-1}} \frac{\partial C}{\partial z_n^L} = \sum_{k=1}^{n} \frac{\partial z_k^L}{\partial a_1^{L-1}} \frac{\partial C}{\partial z_k^L}$$

Ce dernier terme, on l'a calculé précédemment et on a $\frac{\partial C}{\partial z_k^L} = \delta_k^L$. De plus, on sait par l'intermédiaire des dérivées usuelles que $\frac{\partial z_k^L}{\partial a_1^{L-1}} = w_{k1}^L$ Et on obtient donc :

$$\frac{\partial C}{\partial w_{12}^{L-1}} = \partial a_2^{L-2} * \sigma'(z_1^{L-1}) * \sum_{k=1}^{n} w_{k1}^L \delta_k^L$$

On a maintenant tous les éléments nécessaires pour généraliser les équations et obtenir le calcul global :

$$\delta_i^L = \sigma'(z_i^L) * C'(a_i^L)$$

$$\forall l \in [\![1, L-1]\!]; \delta_i^l = \sigma'(z_i^L) * \sum_{j \in \mathbb{N}} w_{ji}^{l+1} \delta_j^{l+1}$$

$$\frac{\partial C}{\partial w_{ij}^l} = a_j^{l-1} * \delta_i^l$$

Dans le cas du biais, on procède de la même manière en tenant compte du fait qu'il n'est pas relié à un neurone d'une couche antérieure ; on peut alors remplacer le terme a_j^{l-1} par 1 et obtenir :

$$\frac{\partial C}{\partial b_i^l} = \delta_i^l$$

1.4.7 Les algorithmes utilisant la rétropropagation

Il existe 5 principales familles d'algorithmes d'optimisation applicables aux réseaux de neurones. La liste des algorithmes est donnée avec un ordre croissant en terme de temps de calcul et de ressources mémoires nécessaires : la descente de gradient, le gradient conjugué, la méthode de Newton, la méthode de Quasi-Newton et la méthode de Levenberg-Marquardt. L'intégralité de ces algorithmes utilise la rétropropagation de l'erreur sous forme de variante mais qui peut être résumée de manière analogue à celle présentée ci-dessus.

Entrées : *données d'entrées, données cibles, nb epochs, taux d'apprentissage*
Sorties : *modèle* $\leftarrow (W, b)$

Initialisation du modèle avec des poids aléatoires
tant que *entrainement non terminé ou nombre epochs non dépassé* **faire**
    pour tous les *Éléments de la liste de données d'entrainement* **faire**
        Calcul de la propagation avant
        Calcul de l'erreur en comparant la cible et le résultat prédit
        Propager l'erreur de couche en couche en arrière
        Mettre à jour tous les poids du réseau
    fin
fin

Algorithme 1.1 : Algorithme d'apprentissage

Les algorithmes d'apprentissage présentés suivent les instructions visibles en 1.1.

Il est nécessaire de signaler que dans ces algorithmes, une itération concerne chaque étape de mise à jour des poids, donc pour chaque échantillon ou batch de données, une itération supplémentaire est comptabilisée. Cependant, une époch est comptabilisée lorsque l'ensemble des échantillons ou batchs de données a été traité. Un batch est un groupe d'échantillon de données. Ils sont utilisés pour faire gagner du temps de calcul à l'algorithme.

1.4.7.1 Descente de gradient

Si l'algorithme de descente de gradient est utilisé pour l'optimisation, les poids seront mis à jour avec la formule :

$$^{k+1}w_{ij}^{l} \leftarrow {}^{k}w_{ij}^{l} - \alpha * \frac{\partial C}{\partial \left({}^{k}w_{ij}^{l}\right)}$$

$$^{k+1}b_{i}^{l} \leftarrow {}^{k}b_{i}^{l} - \alpha * \frac{\partial C}{\partial \left({}^{k}b_{i}^{l}\right)}$$

avec α correspondant au taux d'apprentissage.

Cette méthode est la plus simple mais aussi la plus populaire car très efficace et de moindre complexité. En effet, c'est une méthode du 1er ordre puisqu'elle ne fait appel qu'au calcul de gradient donc aux dérivées partielles du 1er ordre.

1.4.7.2 Méthode de Newton

La méthode de Newton est un algorithme du second ordre car il utilise la matrice Hessienne. L'objectif est de trouver de meilleures directions d'entraînement en utilisant les dérivées secondes.

Les poids sont mis à jour avec l'équation suivante :

$$^{k+1}w_{ij}^{l} \leftarrow {}^{k}w_{ij}^{l} - \alpha * \left(\left(\frac{\partial^2 C}{\partial^2 \left({}^{k}w_{ij}^{l} \right)} \right)^{-1} * \frac{\partial C}{\partial \left({}^{k}w_{ij}^{l} \right)} \right)$$

$$^{k+1}b_{i}^{l} \leftarrow {}^{k}b_{i}^{l} - \alpha * \left(\left(\frac{\partial^2 C}{\partial^2 \left({}^{k}b_{i}^{l} \right)} \right)^{-1} * \frac{\partial C}{\partial \left({}^{k}b_{i}^{l} \right)} \right)$$

La méthode de Newton nécessite moins d'epochs que la descente de gradients pour trouver la valeur minimale de la fonction coût. Cependant, elle a le grand inconvénient de prendre beaucoup de temps pour calculer la Hessienne et son inverse.

1.4.7.3 Le gradient conjugué

Le gradient conjugué est considéré comme une méthode intermédiaire entre la descente de gradient et la méthode de Newton. Elle est motivée par le désir d'accélérer la convergence généralement lente de la descente de gradient et d'éviter les besoins calculatoires de l'évaluation de la matrice Hessienne et de son inverse de la méthode de Newton.

Dans l'algorithme d'apprentissage du gradient conjugué, la recherche est effectuée le long de directions conjuguées obtenues à partir de la seule connaissance des gradients, qui produisent généralement une convergence plus rapide que la descente de gradient.

La mise à jour se fait avec la formule :

$$^{k+1}w_{ij}^{l} \leftarrow {}^{k}w_{ij}^{l} + \alpha * d^{k}$$

Pour simplifier les équations, notons $\frac{\partial C}{\partial \left({}^{k}w_{ij}^{l} \right)} = \nabla C_{w}^{k}$ et $\frac{\partial C}{\partial \left({}^{k}b_{i}^{l} \right)} = \nabla C_{b}^{k}$

On a pour le calcul de la direction de descente :

$$d^{k+1} = -\nabla C_{w}^{k+1} + \gamma^{k+1} * d^{k}$$

avec γ^{k+1} le paramètre conjugué, qui se calcule de différentes manières. Les deux plus utilisées sont la méthode de Fletcher-Reeves et la méthode de Polak-Ribière.

Voici l'expression de γ^{k} avec la 1ère méthode :

$$\gamma^{k+1} = \frac{\left(\nabla C_{w}^{k+1} \right)^{T} \nabla C_{w}^{k+1}}{\left(\nabla C_{w}^{k} \right)^{T} \nabla C_{w}^{k}}$$

et celle avec la seconde méthode :

$$\gamma^{k+1} = \frac{\left(\nabla C_{w}^{k+1} \right)^{T} \left(\nabla C_{w}^{k+1} - \nabla C_{w}^{k} \right)}{\left(\nabla C_{w}^{k} \right)^{T} \nabla C_{w}^{k}}$$

Ce type de méthodes a montré ses preuves en terme d'efficacité durant la phase d'apprentissage des réseaux de neurones car elles n'imposent pas le calcul de la Hessienne.

1.4.7.4 Méthode de quasi-Newton

L'application de la méthode de Newton est très coûteuse en temps de calcul. Elle nécessite de nombreuses opérations pour évaluer la matrice Hessienne et son inverse. D'autres approches, connues sous le nom de méthodes quasi-Newton ont été mises au point pour pallier cet inconvénient. Ces méthodes calculent une approximation de la Hessienne inverse par l'intermédiaire des informations fournies par le gradient uniquement.

La mise à jour se fait avec l'équation suivante :

$$^{k+1}w_{ij}^{l} \leftarrow {}^{k}w_{ij}^{l} - \alpha * (\gamma^{k} * \nabla C_{w}^{k})$$

Pour simplifier les équations, notons $s^k = {}^{k+1}w_{ij}^{l} - {}^{k}w_{ij}^{l}$ et $r^k = \nabla C_w^{k+1} - \nabla C_w^k$

On a pour le calcul d'approximation de la Hessienne, deux principales méthodes utilisées, qui sont celle de Broyden-Fletcher-Goldfarb-Shanno (BFGD) et celle de Davidon-Fletcher-Powell (DFP), la première d'entre elles repose sur l'équation suivante :

$$\gamma^{k+1} = \gamma^{k} + \frac{r^k \left(r^k\right)^T}{\left(r^k\right)^T s^k} - \frac{\gamma^k s^k \left(s^k\right)^T \gamma^k}{\left(s^k\right)^T \gamma^k s^k}$$

et avec la seconde sur une expression d'approximation légèrement différente :

$$\gamma^{k+1} = \gamma^{k} + \frac{s^k \left(s^k\right)^T}{\left(s^k\right)^T r^k} - \frac{\gamma^k r^k \left(r^k\right)^T \gamma^k}{\left(r^k\right)^T \gamma^k r^k}$$

1.4.7.5 Méthode de Levenberg-Marquardt

L'algorithme de Levenberg-Marquardt a été conçu pour fonctionner spécifiquement avec des fonctions coûts qui prennent la forme d'une somme des erreurs au carré. Il propose une amélioration de l'algorithme classique de Gauss-Newton, méthode spécifique de la résolution de problèmes d'estimation de paramètres d'un modèle non linéaire. Il s'appuie sur la forme particulière de la fonction coût comme somme de fonctions au carré, ne nécessitant pas le calcul de la Hessienne mais les seuls calculs du gradient et de la matrice jacobienne.

Considérons une fonction coût qui peut s'exprimer comme la somme des erreurs au carré $C = \sum_{i=1}^{m} e_i^2$ avec m le nombre d'échantillons de données. On peut ensuite définir la matrice Jacobienne de la fonction coût comme celle contenant les dérivées partielles des erreurs par rapport aux paramètres $J_{i,j} = \frac{\partial e_i}{\partial w_j}$ avec $i \in [\![1, m]\!]$ et $j \in [\![1, n]\!]$ où n correspond au nombre de paramètres du réseau de neurones. Le gradient de la fonction

coût peut se calculer avec l'expression $\nabla C = 2J^T \cdot e$ avec e le vecteur d'erreur. Ensuite, on peut approximer la matrice Hessienne HC par l'expression suivante :

$$\mathcal{H}C \approx 2J^T \cdot J + \lambda \mathbb{I}$$

avec λ, un facteur d'amortissement assurant la positivité de la matrice Hessienne, condition nécessaire pour faire diminuer la fonction coût d'une itération à l'autre.

L'expression de mise à jour des poids est :

$$w^{k+1} = w^k - \left(\left(J^k\right)^T \cdot J^k + \lambda^k \mathbb{I}\right)^{-1} \cdot \left(\left(2J^k\right)^T \cdot e^k\right)$$

Lorsque $\lambda = 0$, le calcul correspond à la méthode de Newton, utilisant l'approximation de la matrice Hessienne. Lorsque λ est grand, la méthode devient une descente de gradient avec un faible taux d'apprentissage.

L'algorithme de Levenberg-Marquardt étant une méthode adaptée aux fonctions de type somme des erreurs au carré, cela permet d'être très efficace lors de l'apprentissage. Cependant, il possède quelques inconvénients. Le premier est qu'il ne peut pas être appliqué à des fonctions quadratiques ou d'entropie croisées. De plus, il n'est pas compatible avec les conditions de régularisation. Pour finir, avec de très grands ensembles de données ou de très gros modèles de réseaux de neurones, la matrice Jacobienne devient énorme et nécessite beaucoup de mémoire. Par conséquent, cet algorithme n'est pas recommandé dans ce cas.

1.4.8 L'engouement vers la descente de gradient

La descente de gradient et ses variantes sont les méthodes les plus utilisées de nos jours. En effet, ayant souvent de grandes bases de données, il est nécessaire d'utiliser les méthodes les plus efficaces. Il existe 3 variantes de descentes de gradient, qui diffèrent par la quantité de données utilisées pour calculer le gradient de la fonction coût. En fonction de la quantité de données, un compromis entre l'exactitude de la mise à jour des paramètres et le temps qu'il faut pour l'effectuer est accompli. Ces 3 variantes sont : la descente de gradient par lots, stochastique et par mini-lots. En plus de ces trois variantes, il existe différents optimiseurs mettant à jour différemment les poids.

1.4.8.1 Descente de gradient par lot

Cette méthode implique le calcul des gradients pour l'ensemble des données sur une mise à jour. La descente de gradient par lot peut être très lente et difficile pour les ensembles de données qui ne tiennent pas en mémoire. De plus, elle ne permet pas de mettre à jour le modèle en ligne. La descente de gradient par lot est garantie de converger vers un minimum global pour les problèmes convexes et vers un minimum local pour une fonction coût non convexe.

1.4.8.2 Descente de gradient stochastique

La Descente de Gradient Stochastique (SGD), en revanche effectue une mise à jour des paramètres pour chaque échantillon de données d'apprentissage. La descente de gradient par lot effectue des calculs redondants pour les grands ensembles de données, car elle recalcule les gradients pour des exemples similaires avant chaque mise à jour des paramètres. La SGD supprime cette redondance en effectuant une mise à jour à la fois. Cette méthode est donc généralement beaucoup plus rapide et peut également être utilisée pour l'apprentissage en ligne.

1.4.8.3 Descente de gradient par mini-lot

La descente de gradient par mini-lots est un compromis entre les deux méthodes précédentes avec une mise à jour pour chaque mini-lot. L'avantage de cette méthode est qu'elle réduit la variance des mises à jour des paramètres par rapport à la SGD ce qui peut conduire à une convergence plus stable. Les tailles courantes de mini-lots varient entre 50 et 256, dépendant de l'application. La descente de gradient par mini-lots est généralement l'algorithme de choix, et le terme de SGD est aussi employé lorsque des mini-lots sont utilisés par abus de langage.

1.4.8.4 Optimiseurs spécifiques à la descente de gradient

Plusieurs optimiseurs ont été développés afin de perfectionner l'apprentissage. En effet, il existe la SGD avec momentum [95], gradient accéléré avec la méthode de Nesterov [96], AdaGrad [97], AdaDelta [98], RMSProp, Adam [99], AdaMax, Nadam [100] et AMSgrad. Toutes ces méthodes sont des descentes de gradient avec des mises à jour de poids différentes. Elles utilisent des opérateurs permettant d'améliorer la descente du gradient. L'optimiseur le plus utilisé de nos jours est Adam pour Adaptative moment estimation.

1.4.9 Frameworks existants

Afin de permettre l'utilisation de ces algorithmes au plus grand nombre d'utilisateurs, un certain nombre d'entreprises a récemment rendu open source (accessible à tout le monde) leurs outils. Leurs outils sont souvent complets et optimisés pour le DL. Ils permettent d'utiliser toutes sortes d'algorithmes ou même d'en développer de nouveaux. L'intégralité des outils permettent d'utiliser des graphes statiques, c'est à dire, d'utiliser un nombre d'entrées qui ne varie pas durant l'apprentissage. Seul, pyTorch permet la création de graphes dynamiques. Une liste non exhaustive des frameworks les plus utilisés avec les entreprises de développement associées est donnée :

— Tensorflow développé par Google : sans doute le framework rassemblant la plus grande communauté.
— PyTorch développé par Facebook : beaucoup utilisé en recherche.
— microsoft CogNitive ToolKit (CNTK) : il est connu pour sa compatibilité avec un grand nombre de (GPUs) et sa vitesse de calcul impressionnante.

— MXNet développé par Apache : il est utilisé pour faire de gros projets industriels, cependant il n'est pas énormément documenté.

— Caffe2 développé par l'université de Berkeley : il est connu pour les déploiements mobiles et à grande échelle.

— Keras développé par Google : Il est réputé pour être l'application facile à utiliser (user friendly) de Tensorflow. Cependant, pour l'utiliser il est obligatoire d'avoir Tensorflow d'installé.

— Theano développé par Milla - Institut : il a le privilège d'être le plus ancien des framework et d'avoir été développé par Yoshua Bengio.

De plus, Scikit-learn développé par l'INRIA : il est le framework de référence de machine learning statistique. Il est aussi open source.

Pour finir, les logiciels R (open source) et Matlab® (payant) sont des frameworks généralistes puisqu'ils permettent l'utilisation d'algorithmes de machine learning statistique et de DL.

1.5 Conclusion

On a vu dans ce chapitre, par l'intermédiaire de l'historique, l'incroyable essor qu'a pu connaitre l'apprentissage automatique. On a aussi remarqué que ce domaine a rencontré des revers lui donnant une évolution en dent de scie. Cependant, il est actuellement dans une phase fortement ascendante. Dans ce chapitre, ont aussi été décrits les modèles d'apprentissage automatique utilisés dans ce manuscrit. Ces modèles concernent les FFNNs et les CNNs. De plus, la description des algorithmes d'optimisation utilisée dans la littérature a permis de voir l'importance qu'ils avaient au sein de ces modèles. On a, par exemple, pu voir que les méthodes du second ordre prenaient énormément de temps de calcul, leur empêchant d'être utilisées pour une grande base de données. On a vu que des compromis étaient à faire en fonction de la base de données accessible.

Chapitre 2

Radiothérapie externe et techniques associées

2.1 Introduction

La radiothérapie externe est un type de radiothérapie, appelée ainsi car la source de rayonnements ionisants se trouve à l'extérieur du patient. Ce traitement se fait par l'intermédiaire d'un accélérateur linéaire de particules qui délivre des rayonnements ionisants à très haute énergie de l'ordre de plusieurs MégaélectronVolts (MeV) dont l'objectif est de tuer les cellules cancéreuses.

Les rayonnements ionisants ne faisant pas la distinction entre cellules saines et cellules cancéreuses, il est nécessaire d'avoir une balistique de traitement extrêmement précise afin de préserver au maximum les cellules saines et les organes à risques avoisinant la tumeur. En effet, sur-irradier les tissus sains peut entrainer des effets secondaires indésirables qu'il est recommandé d'éviter au maximum. Afin d'éviter une trop forte exposition aux tissus sains, les recherches en radiobiologie ont montré que les tissus sains se régénéreraient plus rapidement que les tissus cancéreux en fonction de la dose absorbée reçue. Pour cette raison, le traitement est classiquement divisé en fraction de 2 Gray($J.kg^{-1}$), 5 jours par semaine, pendant 6 semaines.

On verra dans ce chapitre que les techniques utilisées ne cessent de s'améliorer au fil des années. Naturellement, ayant des techniques plus élaborées, elles entrainent des procédures et vérifications supplémentaires. Avant de faire le tour des techniques de traitement, une vision d'ensemble des examens et étapes nécessaires de prétraitement est donnée dans la section suivante.

2.2 Vue d'ensemble

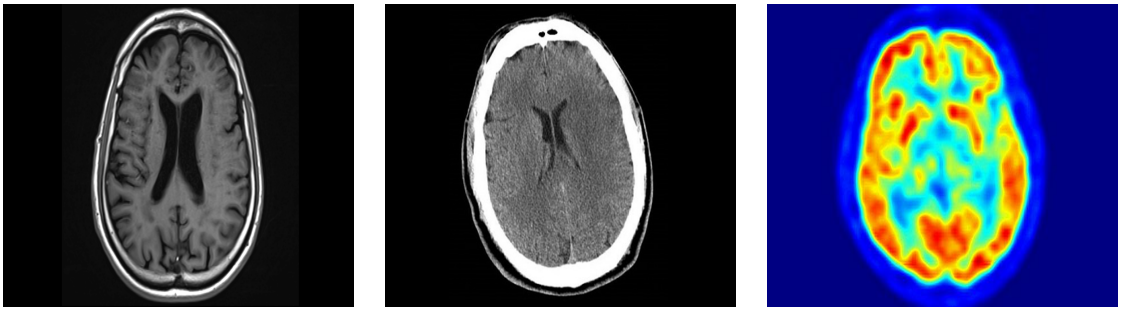
La vision d'ensemble décrite ci-dessous montre les différentes étapes qu'englobent un traitement de radiothérapie externe. En effet, le patient ne va pas arriver le matin et repartir le soir traité. C'est un processus qui prend du temps et qui a besoin de rigueur. Plusieurs étapes constituent un traitement, il est important que l'intégralité de ces étapes soient effectuées chronologiquement.

Pour commencer, le diagnostic du cancer sera établi par un radiothérapeute qui par l'intermédiaire de l'imagerie, pourra détecter s'il y a présence ou non d'un cancer. Une fois le diagnostic posé, les médecins devront contourer les différentes zones à considérer dans les images CT afin que le processus de planification de traitement puisse être mis en place. La phase de planification permet d'établir un plan de traitement personnalisé pour chaque patient en fonction des contours établis. Cette phase précède la phase de dosimétrie qui permettra de valider définitivement le traitement.

Une fois que le traitement est validé par un médecin et un physicien médical, une phase de contrôle qualité est mise en place avant la première séance de traitement. Elle permet de vérifier si la machine est correctement calibrée et délivre bien la séance de traitement planifiée. Si cette phase est correcte, alors les sessions de traitements journaliers et identiques peuvent être effectuées.

2.2.1 Imagerie

L'imagerie a un rôle primordial dans l'étude de l'extension tumorale. Au vu de la complexité de cette maladie, l'information retirée des différentes techniques d'imagerie médicale est essentielle. Ce sont le tomodensitomètre (CT) et l'Imagerie par Résonance Magnétique (IRM) qui sont les plus utilisés à des fins de diagnostic. Ils représentent une imagerie anatomique du patient permettant de détecter des objets anormaux à l'intérieur du patient. De plus, une imagerie fonctionnelle telle que la Tomographie par Émission de Positons (TEP) peut être effectuée. Ces 3 imageries utiles au diagnostic apportent des informations différentes et complémentaires avec un exemple de chacune d'elles exposé en Figure 2.1.



(a) IRM (b) CT (c) TEP

FIGURE 2.1 – Imageries utilisées pour le diagnostic

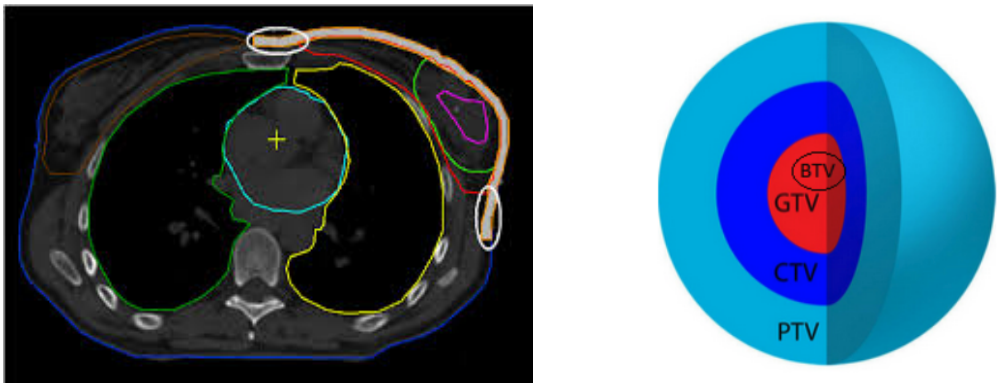
Le Cone-Beam Computed Tomography (CBCT) correspond à un CT de moins bonne résolution. Il est utilisé pour vérifier la position du patient avec une information 3D et s'assurer que les conditions de positionnement soient bien les mêmes que lors de la planification de traitement.

L'EPID est un imageur plan qui était, avant le CBCT, initialement prévu pour la vérification du positionnement du patient. Au fil des années sa fonction a changé pour être utilisée à des fins dosimétriques.

2.2.1.1 Tomodensitométrie

Le premier maillon de la chaîne est le tomodensitomètre (plus communément appelé scanner - CT). C'est de lui que le traitement va dépendre. Sur cette image, vont être établis les différents contours par le médecin comme illustré sur la Figure 2.2a. Le contour de la tumeur mais aussi ceux des différents organes et organes à risques. Pour chacun des objets, différents volumes qui ont été définis par la Commission Internationale des Unités et mesures Radiologiques (ICRU) et plus précisément dans les rapports 50 et 62 [101] seront contourés. Les différents volumes sont visibles sur la Figure 2.2b :

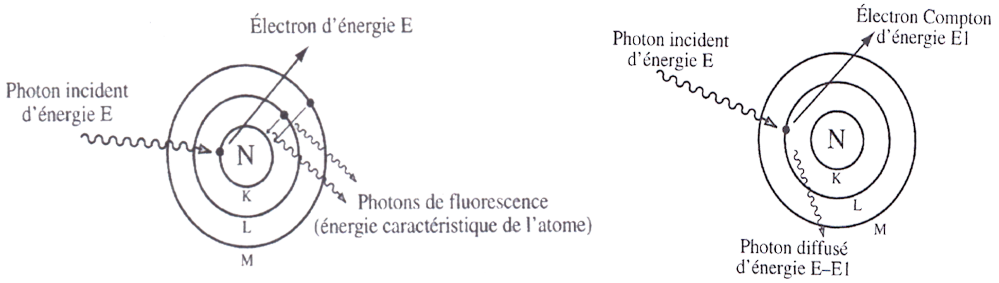
- Le volume tumoral macroscopique (GTV) comprend l'ensemble des lésions tumorales palpables cliniquement et visibles sur l'imagerie.
- Le volume cible anatomo-clinique (CTV) est l'ensemble du volume cible anatomique dans lequel apparaissent les tissus cancéreux visibles macroscopiquement et microscopiquement. Il est défini en prenant des marges autour du GTV afin d'englober les extensions microscopiques de la maladie.
- Le volume cible prévisionnnel (PTV) est défini en prenant des marges de sécurité autour du CTV pour tenir compte des incertitudes de positionnement, des mouvements internes (respiration, remplissage de la vessie...) et des équipements.
- Le volume cible biologique macroscopique (BTV) est peu utilisé. Il est rendu possible grâce à l'apport de l'imagerie fonctionnelle et métabolique.



(a) Contours des organes (b) Différents volumes de contourage

FIGURE 2.2 – Contourages effectués par le médecin

Le tomodensitomètre (CT) est une imagerie fonctionnant à partir de rayons X de faible énergie (entre 40 à 120 keV). La raison principale de l'utilisation de cette énergie est l'apparition prépondérante de l'effet photoélectrique à cette gamme d'énergie permettant un signal net. En effet, l'effet photoélectrique contrairement à l'effet Compton, avec lequel il est en compétition pour une certaine gamme d'énergie, produit l'absorption totale de l'énergie du rayon X incident transmis à l'électron. L'effet Compton quant à lui produit une absorption partielle puisqu'un rayon X diffusé poursuit sa route en plus de l'électron, comme montré sur la Figure 2.3. Ce photon diffusé étant dévié, est à éviter pour l'imagerie puisqu'il va apporter du bruit.



(a) Effet Photo-électrique (b) Compton

FIGURE 2.3 – Phénomènes physiques pour l'imagerie

Les rayons X traversent le corps humain en étant plus ou moins absorbés par les tissus selon leur densité. En effet, plus le tissu est dense tel que de l'os par exemple, plus il absorbera les rayons X et moins il y aura de signal sur le détecteur. La source de rayon X et le détecteur situé à l'opposé, tournent autour du patient durant l'irradiation donnant la reconstruction 3D possible. Les objets (organes,...) situés sur l'image apparaissent dans plusieurs projections avec des aspects différents dépendant de l'angle. La théorie de Radon a établi la possibilité de reconstituer des objets à partir d'un certain nombre de projections. Il existe différentes méthodes de reconstruction telles que les méthodes analytiques et les méthodes itératives. Ces dernières sont aujourd'hui les plus utilisées car elle produisent moins d'artefacts sur l'image, cependant, leur temps de reconstruction est plus lent.

2.2.1.2 CBCT

Le positionnement du patient lors de ses séances d'irradiation successives est essentiel à la qualité du traitement. En effet, si la position n'est pas conforme à celle qui a été planifiée, cela aura un impact sur la DDA délivrée. Le positionnement du patient a logiquement été obtenu pendant de nombreuses années par l'intermédiaire d'un détecteur placé dans le champ d'irradiation, qu'on appelle imageur portal. Logiquement, car cela n'impliquait pas d'irradiations supplémentaires pour le patient. Cet imageur portal subit des faisceaux d'irradiation de plusieurs MégaélectronVolts entraînant une mauvaise qualité d'image avec un mauvais contraste, principalement due au durcissement du faisceau et surtout à l'effet Compton prépondérant à cette énergie. Pour palier ce problème, un imageur CBCT bénéficiant d'un bon contraste, d'une bonne résolution et utilisant des rayons X de quelques dizaines de keV est aujourd'hui utilisé. Il est composé d'un détecteur plan 2D et d'une source située à l'opposé, tous deux positionnés orthogonalement au plan du faisceau de traitement.

La source et le détecteur tournent autour du patient produisant, par l'intermédiaire d'une reconstruction, des images 3D. Cependant la reconstruction du CBCT est un peu différente de celle du CT car le détecteur est plan. Pour un CT, la modalité « fan beam » est utilisée alors que pour le CBCT, c'est la modalité « cone beam ».

En plus de pouvoir l'utiliser pour réaliser le positionnement précis des patients,

il est possible de comparer ces images 3D aux images anatomiques de planification afin d'identifier des modifications anatomiques ou morphologiques au fil des séances de traitement.

2.2.1.3 EPID

L'imageur portal EPID, initialement utilisé pour le positionnement du patient, est placé sous l'appareil de traitement à l'opposé de la tête accélératrice. Ayant été remplacé par le CBCT pour la vérification du positionnement du patient, il trouve de plus en plus son utilité pour des applications à visée dosimétrique. Il n'a pas été fabriqué dans ce but, cependant il possède des qualités intrinsèques qui permettent de le placer comme outil intéressant pour la dosimétrie. En effet, il possède une très bonne résolution spatiale, une rapidité d'acquisition du signal et de transformation en image numérique, une bonne reproductibilité et répétabilité. Mise à part ces qualités, de nombreuses études ont permis de définir des méthodes de correction afin de proposer des solutions dosimétriques abouties. L'EPID est devenu intéressant pour deux principales applications : le CQ (prétraitement) et la dosimétrie *in-vivo* (durant le traitement).

2.2.2 Planification de traitement

Le deuxième maillon de la chaîne concerne la planification de traitement. C'est une pièce maîtresse du traitement. La balistique de ce dernier est élaborée par la mise en place des faisceaux d'irradiation (énergie, angle d'incidence, nombre de faisceaux, nombre de fractions, ...), par le positionnement d'un isocentre (point central du traitement), la Radiographie Digitale Reconstruite (DRR) et la conformation au volume à irradier par l'intermédiaire du placement des mâchoires et du Collimateur Multi-Lames (MLC). A partir des informations décrites par les médecins sur le tomodensitomètre CT, un algorithme d'optimisation va déterminer le plan de traitement qui correspondra au mieux au patient. En effet, en fonction des contraintes posées, l'algorithme permettra de minimiser la dose absorbée dans les organes à risques et de la maximiser dans les tissus cancéreux. L'association américaine de physique médicale (AAPM) a défini une procédure à suivre pour tous les centres cliniques de planification des traitements en radiothérapie externe [102]. Les principales étapes sont récapitulées dans l'encadré de la Figure 2.4.

2.2.3 Dosimétrie

La dosimétrie est la détermination quantitative de la dose absorbée par un organisme ou un objet, autrement dit, c'est la quantité d'énergie reçue par unité de masse, à la suite de l'exposition à des rayonnements ionisants.

Ainsi, le calcul de dose absorbée consiste à comptabiliser l'énergie absorbée en tous points du milieu irradié, où divers processus physiques sont enclenchés en fonction des interactions rayonnements-matière.

1. Positionnement du patient et du système de contention
   - Recherche de la position du patient la plus adaptée au traitement (confort, reproductibilité,...)
   - Recherche du système de contention le plus efficace pour le maintient du patient dans une même position
2. Acquisition des images diagnostiques (CT, IRM et TEP)
   - Acquisition et transfert des images sur le Système de Planification de Traitement (TPS)
   - Recalage éventuel des données (IRM, TEP) sur le CT de référence réalisé en position de traitement
3. Définitions anatomiques
   - Délimitation des contours correspondant aux structures saines, critiques et tumorales avec les différents volumes énoncés auparavant.
   - Obtention de la cartographie des densités électroniques de chaque structure en fonction du contraste des images CT.
4. Définition des faisceaux
   - Détermination des faisceaux (énergie, nombre, incidence,...)
   - Affichage des champs et des structures dans des vues multiples.
   - Conformation des champs aux structures à irradier au moyen de mâchoires et collimateurs multi-lames.
   - Sélection des modificateurs de faisceaux (bolus, filtre en coin, ...).
   - Détermination du ratio de chaque faisceau.
5. Calcul de dose absorbée
   - Sélection de l'algorithme de calcul de dose absorbée, choix de la grille de résolution et calcul de la DDA.
   - Transmission de la dose absorbée prescrite aux physiciens.
6. Évaluation du plan de traitement
   - Analyse visuelle et complémentaire avec des histogrammes doses volumes des distributions de dose absorbée.
   - Utilisation d'outils d'optimisation pour répondre aux objectifs cliniques.
7. Mise en application du plan de traitement
   - Calcul des unité moniteur (relation au temps de traitement par faisceaux)
   - Vérification et validation du plan de traitement.
   - Transfert du plan de traitement vers le système d'enregistrement et de vérification des plans de traitement, puis vers la machine.

FIGURE 2.4 – Procédure de planification

Les phases dosimétriques ont leur importance dans la chaîne puisque ce sont elles qui permettent d'évaluer si le traitement correspond à celui attendu.

Différentes phases dosimétriques interviennent puisqu'il y a la phase de calcul dosimétrique établie par le TPS et la phase dosimétrique de vérification pré et durant le traitement. Ces dernières seront introduites dans le prochain chapitre.

2.2.4 Méthode de calcul

Considérons la dose absorbée dans le patient exposé à un faisceau d'irradiation. Les fermions (electrons, positons,...) vont directement déposer de l'énergie dans le milieu contrairement aux bosons (photons, ...), qui sont décrits comme étant indirectement ionisants. En effet, ils vont transférer leur énergie à des particules directement ionisantes. La précision d'un modèle de calcul de dose absorbée dépend ainsi de sa capacité à reproduire et à quantifier les phénomènes physiques d'interaction avec le milieu (en prenant en compte les coefficients d'atténuation des différents milieux, leur section efficace et les phénomènes de transports d'énergie).

Le calcul de la DDA par le TPS est primordial et doit être précis. Il a été constaté ces dernières années, une évolution considérable des algorithmes implémentés dans les TPSs. Cela est dû aux évolutions des systèmes d'imagerie et à la puissance calculatoire des ordinateurs. Ces prouesses technologiques diminuent considérablement le temps de calcul et permettent une utilisation clinique.

À l'heure actuelle, il y a deux principales méthodes de calcul concernant le calcul dosimétrique établi par le TPS : les méthodes analytiques et les méthodes de Monte-Carlo (MC).

2.2.4.1 Méthodes analytiques

Les méthodes analytiques sont basées sur des modèles de calculs. Modèles qui ont régulièrement été alimentés par de nouvelles innovations, mais dont les principes de base sont restés. C'est essentiellement les nouveautés de l'informatique qui ont conditionné les versions successives des logiciels. Parmi les principaux modèles analytiques figurent les modèles empiriques, les modèles de superpositions, les modèles de convolutions/superpositions et le modèle AAA.

Les modèles empiriques sont les plus anciens et furent initialement pratiqués manuellement. Un certain nombre de faisceaux mesurés dans des conditions de références étaient enregistrés dans des tables. Les valeurs situées entre les données de base étaient interpolées. Le modèle empirique a eu des évolutions avec sa repésentation analytique utilisant différentes corrections telles que la prise en compte des filtres en coin ou bien des hétérogénéités tissulaires. Plusieurs concepts tels que la profondeur équivalente, la correction à partir des rendements en profondeur, la correction Power law (BATHO) [103], la correction par soustraction du faisceau [104] ou la correction rapport tissus-air équivalent [105] permettent de prendre en compte les hétérogénéités tissulaires.

Les méthodes par superposition consistent à superposer les contributions provenant de différents éléments couvrant l'ensemble du volume irradié. Ce principe est appliqué avec plusieurs méthodes telles que la séparation primaire diffusée, l'utilisation de point kernel ou la méthode *Pencil Beam*.

Le principe de séparation de la contribution du rayonnement primaire et du rayonnement diffusé a été introduit en 1972 [106]. Ses travaux se sont appuyés sur la méthode de Clarkson [107] qui décomposait la contribution du rayonnement diffusé par secteurs angulaires.

Les points kernels permettent de prendre en considération de manière efficace le transport des particules secondaires. Cette méthode décrit la manière dont l'énergie est déposée dans un milieu (en général de l'eau) autour d'un site d'interaction d'un photon primaire mono-énergétique. De nos jours, le point kernel est couramment calculé par simulation Monte-Carlo [108, 109].

Le pencil beam se base sur des points kernels mesurés. Ils sont déterminés à plusieurs profondeurs pour chaque énergie. Le calcul de la dose absorbée à d'autres profondeurs se fait via des interpolations entre les points kernels pré-calculés. Dans le cas de la présence d'héterogénéités, l'algorithme utilise par exemple la loi de Batho modifiée en ne tenant compte que de l'atténuation du faisceau. Avec cet algorithme l'influence latérale des hétérogénéités n'est pas prise en compte.

Les méthodes de convolution/superposition utilisent la méthode de superposition ainsi que l'application de produit de convolution. La méthode Collapsed Cone convolution [110] est la plus utilisée des méthodes à convolution/superposition. Elle offre un bon compromis temps/précision. Elle utilise des points kernels exprimés en coordonnées sphériques, permettant d'avoir une direction de transport d'énergie de forme conique en partant du point central.

Le modèle AAA est la méthode utilisée dans la plupart des TPSs Eclipse de la machine Varian®. C'est un algorithme complet tenant compte des hétérogénéités latérales. Un des inconvénients concerne son manque de précision dans le calcul pour les organes de très faible densité tels que les poumons [111].

2.2.4.2 Méthodes Monte-Carlo

La méthode de calcul MC est connue pour être le *gold standard* (standard de référence) en physique médicale. En effet, elle est connue pour sa précision [112, 113]. Précision qu'elle acquiert par la simulation probabiliste du transport des particules et de leurs interactions avec la matière. Elle estime correctement la distribution spatiale des particules et leurs dépôts d'énergie. Elle utilise un générateur de nombres aléatoires afin de simuler individuellement les trajectoires et dépôts d'énergie de chaque particule. Leur histoire est suivie jusqu'à leur épuisement en terme d'énergie.

Les distributions probabilistes sont intelligemment échantillonnées en fonction des différentes sections efficaces des matériaux traversés et de l'énergie des particules.

Le grand point négatif de cette méthode est un temps de calcul conséquent. En effet, simuler l'intégralité des processus puis suivre l'évolution de chaque particule demandent de grandes ressources informatiques.

De nos jours, certains algorithmes de TPS incluent les méthodes Monte-Carlo en approximant, via différentes techniques de réduction de variance notamment, le calcul. Cela permet d'avoir un temps de calcul convenable nécessaire pour la routine clinique.

Plusieurs codes de calculs Monte-Carlo peuvent être utilisés dans le domaine de la physique médicale et de la recherche. Les principaux sont EGS (Electron Gamma Shower) [114], BEAMnrc [115], Penelope (PENetration and EnergyLOss of Positrons and Electrons) [116], MCNP (Monte Carlo N-Particle transport) [117], Geant4 (GEometry And Tracking) [118] et Gate (Geant4 Application for Tomographic Emission) [119, 120].

2.2.5 Contrôle qualité

Le contrôle qualité du traitement est une partie importante du traitement afin d'éviter des conséquences dramatiques. En effet, les résultats thérapeutiques sont liés au respect des protocoles mis en place par les physiciens permettant de contrôler la bonne réalisation du traitement. Chaque étape exposées précédemment doit être analysée afin de minimiser les sources d'erreurs d'origine technique et pouvoir garantir la cohérence du traitement planifié par les professionnels de santé.

Dans ce cadre là, différents contrôles qualité sont effectués. Ils portent particulièrement sur la précision mécanique des éléments de collimation du faisceau, la constance de la qualité des faisceaux de rayons X ou électrons, sur les éléments d'imagerie que sont les imageurs portals EPID et systèmes d'imageries embarqués (CBCT). La constance de transfert des paramètres du traitement sur l'accélérateur sont également vérifiés. Par exemple, les débits de tous les faisceaux d'irradiations des accélérateurs sont quotidiennement contrôlés. Mais il y a aussi des contrôles qualité concernant le calcul de dose absorbée des traitements et la réalisation de ces derniers.

Lors de l'installation d'un nouvel accélérateur, le radiophysicien réalise un panel de mesures de dose absorbée permettant de modéliser les traitements selon différentes modalités. Il est de sa responsabilité de s'assurer de la justesse des calculs réalisés. Pour cela, les doses absorbées restituées par le TPS sont comparées aux doses absorbées mesurées. Des tests réalisés dans plusieurs configurations simples permettent de valider la majorité des traitements. La prise en compte des hétérogénéités tissulaires est étudiée en utilisant des fantômes anthropomorphiques, simulant le corps humain. Dans le domaine médical, sont appelés fantômes, des objets permettant de simuler une présence de patient sur la table de traitement. Ces objets peuvent avoir différentes formes et densités allant de la cuve d'eau au fantôme anthropomorphique.

Des simulations de traitements sont réalisées en contrôlant que tous les paramètres mis en place lors de la planification de traitement soient transmis correctement au poste de traitement. Le calcul du nombre d'Unités Moniteur (UM où 1 UM=1 cGy dans les conditions de références) déterminant la dose absorbée délivrée par séance est vérifié.

Pour finir, une vérification de tous les faisceaux d'irradiation est effectuée lors de la première séance préalablement au traitement. Cette dernière permet de voir si la dose absorbée délivrée par l'accélérateur est conforme à celle planifiée. L'imageur portal EPID est souvent utilisé pour cette phase de vérification car c'est le seul équipement qui peut apporter un retour d'informations durant le traitement. Étant placé dans l'axe du faisceau, il contient une information temps-réel du traitement délivré.

Différentes approches permettent de contrôler la qualité du traitement délivré par l'accélérateur avant la première séance. La première consiste à placer un fantôme d'eau sur la table et de mettre l'EPID dans les mêmes conditions que durant le traitement. La deuxième consiste à placer l'EPID à l'isocentre du traitement. Toutes deux ont un inconvénient correspondant dans le premier cas, à la logistique contraignante de placer une cuve d'eau sur la table, puis dans le second cas, de mettre l'EPID dans des conditions de mesures différentes au traitement. La troisième consiste à placer l'EPID dans les mêmes conditions que durant le traitement sans rien placer sur la table de traitement. C'est cette approche qui a été choisie lors des travaux exposés dans ce manuscrit. On a pu citer un certain nombre de contrôles qualités en radiothérapie externe non exhaustif. Dans la suite de ce manuscrit, lorsque l'on fera référence au contrôle qualité, il s'agira de la dernière phase énoncée, concernant la vérification précédent la première séance de traitement.

2.3 Techniques de traitement

Avant d'énoncer les différentes techniques de traitement, il est nécessaire d'introduire la machine permettant de traiter les patients. Il existe un certain nombre de modèles différents qui sont développés par des entreprises. Ces machines sont des accélérateurs linéaires de particules (LINAC) tels que montrés sur la Figure 2.5. Ils font appel à des technologies sophistiquées et ont plusieurs modes de fonctionnement. Le principe physique sous-jacent est l'accélération des électrons provenant du canon à électrons par l'intermédiaire d'ondes électromagnétiques à hautes fréquences.

Deux modes de traitement sont possibles. Soit le traitement par électrons sortis du canon puis accélérés, ce mode est peu utilisé car il permet de traiter des tumeurs en contact de la peau ou très peu profondes. Soit par des photons, mode le plus utilisé, il est acquis par l'intermédiaire des électrons convertis par interactions (principalement rayonnement de freinage) sur une cible de tungstène. Les photons obtenus sont de haute énergie et peuvent pénétrer en profondeur pour des tumeurs localisées dans le patient. L'énergie diffère en fonction du traitement, les machines fournissent généralement des faisceaux d'irradiation de 6 MeV à 25 MeV. Par abus de langage, le terme de MégaVolts est utilisé par les constructeurs mais cela correspond à une énergie en MégaélectronVolts.

Le LINAC est composé de deux principales parties : la section accélératrice et la tête de l'accélérateur de particules. Le premier élément situé à l'entrée de la section accélératrice est le canon à électrons qui permet l'émission d'électrons produits par effet thermoélectronique via une plaque de tungstène chauffée par un filament spiralé. Dès

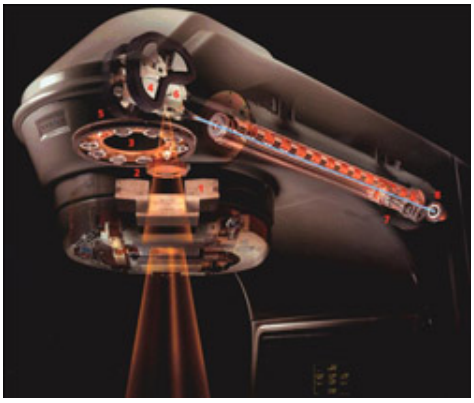


FIGURE 2.5 – Vision intérieure de la tête et de la section accélératrice du LINAC

lors, ces électrons sont dirigés vers une cavité de regroupement avant de traverser la section accélératrice. Cette dernière permet d'accélérer les électrons à chaque passage dans les différentes cavités délivrant une tension positive croissante. Les électrons obtiennent alors, une énergie cinétique importante avant d'interagir avec la cible. Les accélérations sont possibles grâce au klystron aussi appelé magnétron qui produit les ondes électromagnétiques hautes fréquences.

Cette section est placée sous vide à forte pression afin de propager les électrons dans chaque cavité. Le faisceau d'électrons arrivant horizontalement, il est nécessaire de dévier sa trajectoire de 270 ° avant d'interagir avec la cible. Cette cible est située en amont de la tête de l'accélérateur de particules et permet la production de rayon X. Une fois que les rayons X sont produits, ils passent dans la seconde partie, la tête de l'accélérateur. Elle permet de mettre en forme le faisceau. Pour cela, plusieurs éléments sont nécessaires tels que le collimateur primaire, le cône égalisateur, la chambre d'ionisation, le filtre en coin et le collimateur secondaire. Ce dernier est composé de mâchoires et du MLC pouvant effectuer un mouvement de rotation dans le but de mieux se conformer à la tumeur.

Le collimateur primaire est une pièce rectangulaire creusée d'un cône. Grâce à cette forme, il limite la direction des rayons X et laisse passer uniquement ceux qui partent vers le patient. Le cône égalisateur permet d'obtenir une DDA plate à une certaine distance qu'on appelle Distance Source-Peau (DSP) de 100 cm de la tête accélératrice. Le cône égalisateur est une option, par exemple, dans les traitements stéréotaxiques, il n'est pas utilisé.

Les traitements stéréotaxiques ne font pas partie de cette étude, il consiste à envoyer un fort débit de dose sur peu de séances de traitement. Cela a le principal avantage de faire gagner un temps considérable de temps de traitement, très recherché en clinique dû à un nombre accru de patients à traiter. Une double chambre d'ionisation placée au-dessus des mâchoires, mesure la dose absorbée par transmission et permet l'asservissement de la valeur du débit de dose. Elle est constituée d'une cavité fermée lui permettant de ne pas répondre en fonction de la température et de la pression. Le contrôle se fait en Unité Moniteur (UM). Généralement, une UM correspond à un cGy

dans les conditions de références. Les conditions de références sont indiquées dans un protocole d'étalonnage suivi par les physiciens.

Ensuite, le filtre en coin est lui aussi optionnel. Il est utilisé pour adapter la DDA à la morphologie du patient et permet d'incliner les isodoses. Pour finir, le collimateur secondaire permet de se conformer au volume à traiter grâce aux mâchoires et au MLC. Les mâchoires délimitent le champ d'irradiation, par une forme carré ou rectangulaire. Elles sont composées de 4 blocs. Le MLC permet lui, de se conformer de manière précise aux contours du volume à traiter.

Plusieurs techniques de traitement se sont développées ces dernières années apportant chacune des évolutions sur les traitements précédents. La capacité des machines et les améliorations algorithmiques ont permis un gain considérable en termes de qualité et précision des traitements. Quelques unes d'entre elles sont citées par la suite par ordre chronologique d'apparition.

(a) Tête de l'accélérateur de particules (b) Collimateur multi-lames

FIGURE 2.6 – Image représentative du MLC

2.3.1 Radiothérapie conformationnelle

La radiothérapie conformationnelle a fait son apparition dans les années 1990. L'objectif de ce traitement est de conformer le faisceau d'irradiation à la forme de la tumeur (cible), en épargnant un maximum de tissus sains. Afin de remplir cet objectif, des blocs composés d'alliage de métaux à base de plomb appelés mâchoires et, plus récemment des MLCs, permettent d'atténuer fortement le faisceau d'irradiation. La Figure 2.6b est représentative du MLC avec un montage de 60 lames de chaque coté de la tête de l'accélérateur de particules pouvant être déplacées de manière latérale.

Certaines nouvelles machines possèdent deux étages de collimateurs multi-lames montés en quinconce afin d'éviter le problème de fuites interlames. En effet, les lames pouvant être déplacées latéralement, il existe un petit espace entre chacune d'entre elles pouvant laisser passer quelques rayons X. Dans ce type de traitement, les lames sont

déplacées avant le traitement et restent statiques durant l'irradiation.

2.3.2 Radiothérapie conformationnelle à modulation d'intensité

Si la RC permettait de conformer les isodoses à la forme de la tumeur, elle irradiait de manière quasi-homogène en terme de DDA. La RCMI (ou IMRT en anglais), est une version de traitement évoluée par rapport à la RC puisqu'elle consiste à délivrer une DDA avec une séquence dynamique des positions des lames. Cette séquence dynamique peut être rythmée de différentes façons. Soit la séquence est discrétisée et les lames restent statiques durant l'irradiation, on l'appelle la méthode « Step and Shot ». Soit la séquence est continue et les lames sont dynamiques durant l'irradiation, on l'appelle la méthode « sliding window » ou « dynamic IMRT ». Dans les deux cas, la RCMI conduit à une distribution hétérogène de dose absorbée. Ce type de traitement est délivré par le TPS après un calcul d'optimisation permettant de déterminer la DDA la plus cohérente tout en respectant les contraintes posées. Habituellement, la RCMI est réalisée par plusieurs faisceaux d'irradiation (généralement entre 5 et 9).

2.3.3 Arc-thérapie volumique modulée

Le VMAT correspond à une évolution de la RCMI, puisqu'elle implique la rotation du bras de l'accélérateur de particules. Cette technique délivre le traitement par l'intermédiaire de deux arcs, il permet de se conformer à la cible en 3D. Ce type de traitement apporte un avantage considérable comme montré sur la Figure 2.7. Il évite de sur-irradier les tissus sains par multiplication des points d'entrées. En effet, le centre du traitement, correspondant à la tumeur, reçoit la somme des faisceaux d'irradiation de chaque angle alors que, plus on s'éloigne de cet isocentre, moins les tissus sont irradiés. La planification de ce type de traitement se fait via le TPS qui va utiliser des algorithmes d'optimisation sophistiqués permettant en fonction des contraintes 3D posées, de trouver le traitement le plus pertinent pour chacun des patients.

2.4 Conclusion

La radiothérapie externe est l'application directe de ce travail. Dans ce chapitre, il a été énoncé les différentes étapes nécessaires pour ce type de traitement. On a vu que l'imagerie avait toute son importance de nos jours afin de pouvoir diagnostiquer et traiter au mieux la maladie. De plus, le système de planification de traitement utilise des algorithmes d'optimisation qui calculent le plan en fonction des contraintes posées sur les images tomodensitométriques CT par les médecins.

La dosimétrie est calculée en fonction du plan précédemment établi permettant aux physiciens d'évaluer si ce dernier peut être retenu. Pour finir, les différents types de traitement existants ont été exposés. Leur complexité ne cesse d'augmenter, c'est

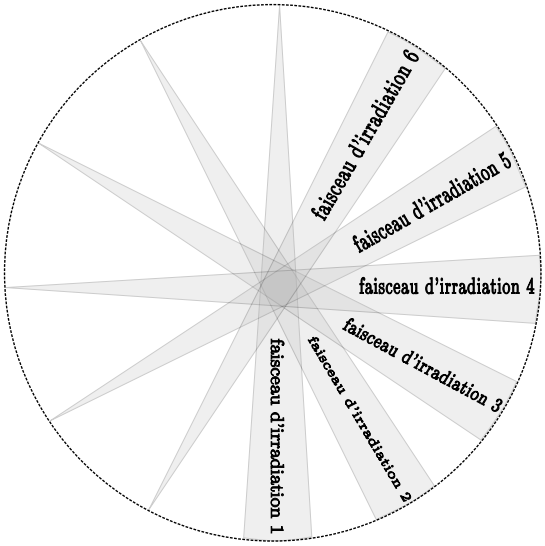


FIGURE 2.7 – Technique d'irradiation en VMAT avec plusieurs points d'entrées

pour cette principale raison que la vérification durant chaque séance de traitement est primordiale.

Chapitre 3

L'EPID pour la dosimétrie

3.1 Introduction

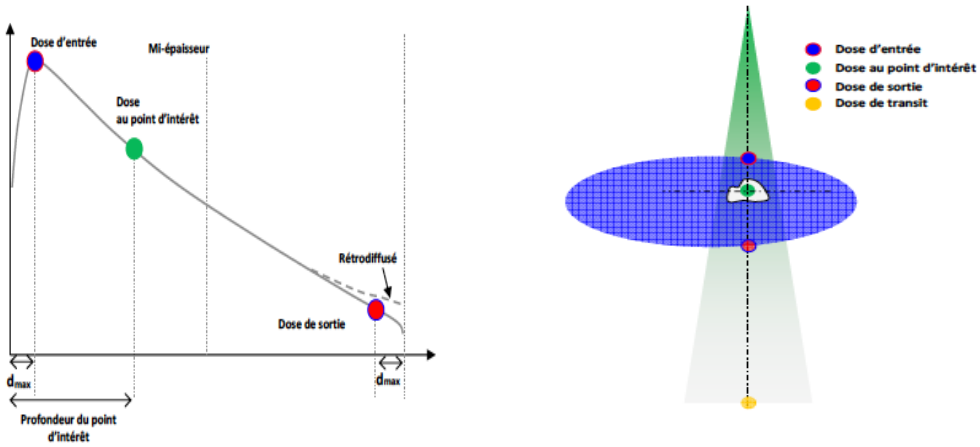
Il a été mentionné dans le précédent chapitre que les traitements sont d'une complexité croissante avec un panel de paramètres et de degrés de liberté possibles de la machine à prendre en considération. En effet, on a vu que pour les traitements VMAT, un certain nombre d'équipements pouvaient être en mouvement : les lames du collimateur qui peuvent avoir un mouvement latéral, le collimateur lui même qui peut avoir un mouvement de rotation et le bras de l'accélérateur qui tourne autour du patient. Toutes ces possibilités entrainent une meilleure conformation à la forme de la tumeur en 3D. Cependant, cela nécessite une étape de vérification précise de la dose absorbée réellement reçue par le patient qui sera comparée à la DDA prescrite.

L'imageur portal peut aujourd'hui répondre à cette question de vérification puisqu'il peut être utilisé à des fins dosimétriques. Plusieurs domaines de recherche ont éclos ces dernières années le concernant. Dans ce chapitre, des notions de dosimétrie avec les différents points de mesure seront introduites. Par la suite, le sujet portera sur le détecteur utilisé au cours de cette thèse, l'EPID. Ses caractéristiques ainsi que la méthode d'acquisition des images seront énoncées. De plus, les différentes méthodes de calculs dosimétriques basées sur l'EPID ainsi que la nouvelle approche exposée au cours de ce manuscrit seront décrites.

3.2 Notions de dosimétrie

La dosimétrie se définit comme la métrologie et la modélisation de l'énergie associée aux radiations ionisantes. D'une manière générale, la dosimétrie consiste à mesurer les doses absorbée reçues par les personnes exposées aux rayonnements ionisants. L'énergie transportée par les photons dirigés vers le patient est à l'origine de la dose absorbée délivrée, et par conséquent des effets biologiques ultérieurs. De ce fait, pour l'application thérapeutique des rayonnements, il est indispensable de comprendre comment la dose absorbée se répartit dans la matière et quels sont les paramètres pouvant influencer la répartition. La mesure de dose absorbée se fait par l'intermédiaire de détecteurs placés à différents points de mesure. Certains détecteurs sont positionnés directement sur le patient en entrée ou en sortie du faisceau. La dose absorbée mesurée durant le

traitement peut alors être comparée à la valeur théorique attendue et calculée par le TPS. Les principales possibilités de mesure dans la pratique sont montrées en Figure 3.1 et sont les suivantes :



(a) Visualisation sur une courbe de rendement en profondeur

(b) Visualisation sur une coupe transversale

FIGURE 3.1 – Représentation des principaux points de mesures pour la DIV [121]

— La dose absorbée en entrée

La dose absorbée en entrée est une mesure utile pour vérifier si la machine émet le bon nombre d'unité moniteur ou bien si la distance source-peau est respectée. Elle peut aussi être utilisée pour vérifier la présence d'accessoires ou le calcul du temps d'irradiation. La dose absorbée à l'entrée mesurée peut directement être comparée à celle planifiée. En cas d'erreur, les physiciens médicaux interviennent en fonction de ce qui a préalablement été défini dans la procédure d'assurance qualité du service de radiothérapie. Les détecteurs utilisés pour la mesure de dose absorbée à l'entrée ont une épaisseur sensible de 1 mm. Cela peut être problématique, d'autant plus que le gradient de dose absorbée engendré par l'équilibre électronique qu'on appelle *BuildUp* peut altérer la précision de la mesure. L'équilibre électronique est respecté à une profodeur où l'influence du diffusé provenant des alentours du point de mesure est identique.

De plus, les électrons de contamination provenant de la tête de l'accélérateur peuvent aussi modifier la mesure. Pour palier ces problèmes, il existe des capuchons de dimension et de densité adaptés venant recouvrir le volume sensible du détecteur. Cependant, la réalisation d'une mesure dans cette configuration engendrera une modification de la dose absorbée déposée en aval du détecteur consécutivement à l'atténuation du faisceau [122]. Lorsque la mesure est faite uniquement lors de la première séance, cette configuration n'aura pas de gros impact sur la dose absorbée déposée dans le patient. Cependant, si la vérification est faite durant chaque séance, elle pourrait engendrer un sous dosage conséquent sur la dose absorbée totale.

— La dose absorbée en sortie

Elle considère l'anatomie du patient, autrement dit, l'épaisseur et les hétérogénéités de densité traversées. En sortie du patient, une zone d'équilibre électronique appelée *BuilDown*, par analogie à la zone de *BuildUp*, subsiste. Elle est due au manque de rétrodiffusion impliqué par l'air qui a une faible densité en sortie du patient. Suite à cette problématique, la position à laquelle la dose absorbée en sortie doit être définie est plus difficile que celle de la dose absorbée en entrée. Habituellement, la mesure de dose absorbée en sortie est effectuée à distance maximum (distance à laquelle la distribution de dose absorbée est maximum) par rapport à la sortie du patient. Pour des raisons pratiques, un capuchon d'équilibre électronique est utilisé en admettant une erreur de mesure car la zone d'influence du capuchon est supérieure à la distance maximum.

— La dose absorbée dans le volume cible

La dose absorbée dans le volume cible peut s'obtenir de différentes manières selon les conditions dans lesquelles on se retrouve. La première est de placer un détecteur directement à l'intérieur d'un « fantôme » par exemple. En physique médicale, on appelle « fantôme » un objet que l'on place sur la table pour remplacer le patient. Cet objet peut être de différentes formes selon les attentes des physiciens. Par exemple, cela peut être un fantôme d'eau, souvent utilisé pour contrôler la qualité du faisceau, ou alors un fantôme anthropomorphique pour simuler l'irradiation d'un patient réel. La seconde est de combiner les doses absorbée mesurées en entrée et en sortie du patient pour estimer la dose absorbée dans le volume cible. Des méthodes consistent à émettre une hypothèse de symétrie au niveau de l'épaisseur traversée dans le patient [123-125]. La plupart des régions anatomiques sont appropriées pour l'utilisation de cette méthode sur l'axe sagittal. Cependant, cette méthode n'est pas applicable pour des faisceaux antéro-postérieurs car sur l'axe coronal, la symétrie n'est pas respectée. La troisième possibilité est de placer directement un détecteur dans le patient [126, 127]. Cette dernière est une méthode intrusive qui reste difficile à appliquer.

— La dose absorbée de transit

Plus récemment, l'utilisation de la dose absorbée de transit déterminée par un imageur portal à haute énergie est devenue un axe majeur de recherche en raison de la grande disponibilité des EPIDs dans les services de radiothérapie. Une hypothèse faite pour sa réalisation est que la mesure obtenue en sortie du patient contienne toute l'information dosimétrique nécessaire au contrôle de la séance de traitement. Á aujourd'hui, deux approches sont retenues. La première est l'approche directe qui consiste à comparer, au niveau du détecteur, l'image EPID avec un algorithme ayant simulé la réponse que l'on aurait dû obtenir. La seconde est l'approche indirecte qui consiste à convertir l'image en dose absorbée au niveau du détecteur avant de la "rétroprojeter" dans le patient. L'avantage considérable de cette dernière (plus difficile à obtenir) est la possibilité d'avoir une DDA au point d'intérêt. Dans ce cas, le résultat obtenu pourra être comparé au calcul du TPS.

3.3 Imagerie portale EPID

L'EPID est un imageur portal situé à l'opposé de la tête de l'accélérateur comme montré sur la Figure 3.2. Il permet d'acquérir des images numériques durant une irradiation. Il est composé d'une matrice active de pixels contenant un signal analogique en fonction de l'irradiation reçue et d'un circuit électronique permettant la conversion en niveau de gris (signal numérique). Il est apparu dans les années 1980, initialement pour vérifier la position du patient sur la table durant le traitement. Historiquement, l'imagerie portale était réalisée par l'utilisation de cassettes contenant un film radiographique. Il s'est avéré que ce fonctionnement avait des inconvénients, surtout sur le plan pratique. Cela prenait un certain temps de retirer le film de la cassette avant de le faire développer.

Pour ces raisons de praticité, 3 principaux EPIDs ont été commercialisés en raison de leur facilité d'utilisation en routine clinique :

— EPID avec caméra CCD
— EPID à chambre d'ionisation
— EPID au Silicium Amorphe (a-Si)

C'est ce denier qui sera retenu par la suite dans cette étude.

Durant cette étude, le choix d'utiliser différents EPIDs a été fait. Pour cela, la contribution de plusieurs centres cliniques a été nécessaire. Chaque centre possède généralement des modèles de même marque afin de se fidéliser auprès du constructeur. La participation de l'Institut Universitaire de Cancérologie de Toulouse (IUCT) et la clinique Pasteur de Toulouse ont permis d'utiliser des accélérateurs de particules de la marque Varian® visible dans la Figure 3.2a. De plus, une collaboration étroite a été entreprise avec l'Institut de Cancérologie de Bourgogne (ICB) permettant de tester les algorithmes sur des accélérateurs de particules de marque Elekta®) montré en Figure 3.2b. Ces deux marques sont les plus reconnues dans le domaine de la radiothérapie externe et se partagent les centres cliniques. Ils proposent plusieurs modèles capables de traiter avec les différentes techniques de traitement existantes.

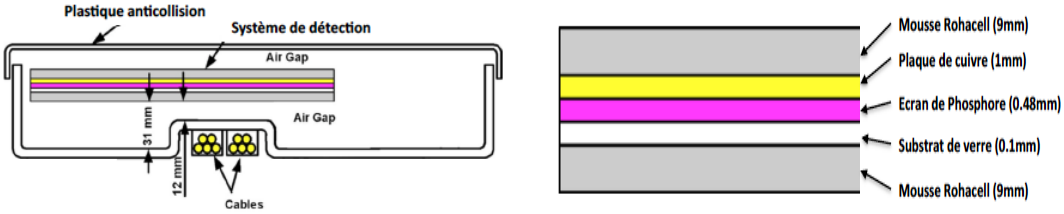
3.3.1 Propriétés physiques

Il a été commercialisé au début des années 2000 pour la première fois après de nombreuses recherches et collaborations scientifiques [128-130]. Cet EPID est, comme on peut le voir sur la Figure 3.3, composé :

— D'une plaque de cuivre de 1 mm d'épaisseur
— D'un écran de phosphore à oxysulfure de gadolinium
— D'une matrice de pixels constituée de photodiodes reposant sur un substrat de verre
— Des mousses rigides (Rohacell) d'imide polyméthacrylique de 9 mm d'épaisseur qui assurent la protection du dispositif de détection
— D'un plastique anticollision de 1,6 mm d'épaisseur avec une distance d'environ 1,5 cm entre le plastique et la surface du détecteur

(a) Varian® Clinac (b) Elekta® Synergy

FIGURE 3.2 – LINAC



(a) Schéma d'une coupe sagittale d'un EPID (b) Zoom sur les couches de matériaux composant l'EPID

FIGURE 3.3 – Composition physique de l'EPID [131]

Lorsque les photons ont réussi à traverser le patient, certains d'entre eux vont être confrontés à la première couche de l'EPID qui est la couche de cuivre, comme montré sur la Figure 3.4. Elle permet de protéger le système d'imagerie des rayonnements diffusés.

Les photons incidents entraînent majoritairement une production d'électrons Compton et des photons diffusés. Les électrons créés et les photons primaires vont interagir en dessous avec un scintillateur. Le scintillateur correspond à un écran de phosphore émettant des photons par luminescence. La lumière qui s'en échappe arrive directement sur la matrice de photodiodes. Afin d'améliorer l'efficacité du détecteur, une mince feuille de métal est placée en amont du scintillateur permettant de convertir plus de photons incidents en électrons. Cela entraine l'augmentation du signal dans les photodiodes ainsi que la sensibilité du détecteur [132].

Chaque pixel de la matrice possède une diode, qui transforme les photons issus de la couche de phosphore en charge et l'accumule jusqu'à la lecture. Chaque fois qu'un photon optique arrive au niveau de la photodiode, une paire électron-trou est créée. Lors d'une acquisition d'image, le commutateur de la photodiode reste non conducteur, de manière à ce que la charge générée indirectement par l'interaction des rayons X avec

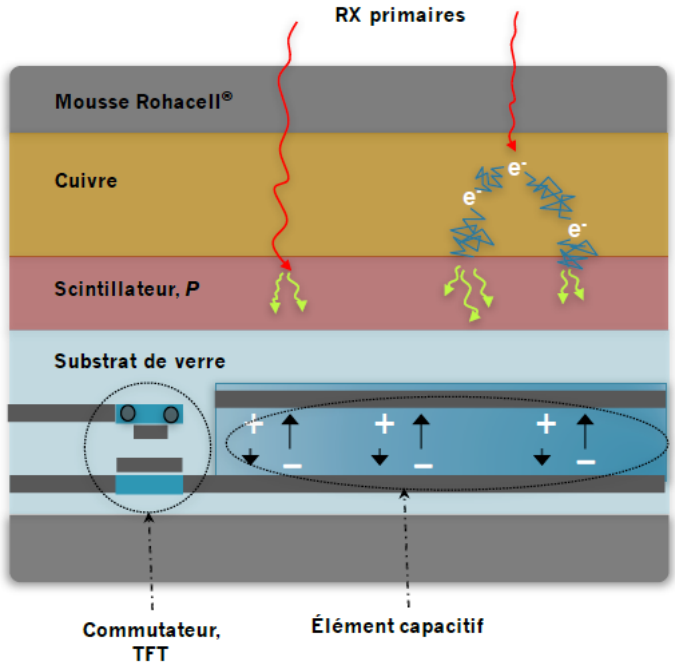


FIGURE 3.4 – Collimateur multi-lames

le matériel de conversion soit stockée dans l'élément capacitif. La lecture se fera au moment où le transistor TFT sera rendu conducteur. Le signal électrique est alors lu et codé sous forme d'image numérique.

3.3.2 Acquisition

Le signal généré par l'EPID est lu et implémenté numériquement. La lecture du signal se fait ligne par ligne. Une image appelée « trame » (ou *frame* en anglais), se forme lorsque toutes les lignes ont été lues. Deux modes d'acquisition des images sont utilisés en clinique. Un mode intégré, où les trames enregistrées pendant l'acquisition sont assemblées pour obtenir une image finale. Cependant, dans ce cas, il est nécessaire de multiplier l'image finale qui représente la moyenne de l'ensemble des trames par le nombre de trames acquises afin d'obtenir un signal proportionnel à la dose absorbée.

Attention, sa réponse sera linéaire si le signal du TPS a été auparavant multiplié par le facteur « DoseGridScaling » présent dans l'en-tête Digital Imaging and COmmunications in Medicine (DICOM) expliqué en 4.2.1.1. Un mode continu [133] (cine pour Varian®) et movie pour Elekta®) où le résultat est un film, donc une succession d'images, représentant l'évolution du traitement. Si ce mode est choisi durant un traitement VMAT, l'ensemble des trames acquises possèdent différents angles d'irradiation puisque le bras de l'accélérateur tourne autour du patient durant le traitement. Il convient de noter que les modes d'acquisition continus actuellement disponibles ne sont pas conçus pour des applications dosimétriques comme objectif principal. Cela implique des options exposées par les fabricants proposant des acquisitions qui diffèrent selon la marque et

le modèle de l'accélérateur de particules.

Pour le modèle Varian® Clinac, l'acquisition clinique ne renvoie pas l'ensemble des trames acquises mais plutôt un sous ensemble d'images moyennant des séries de trames. Le nombre de trames par image peut être contrôlé par l'utilisateur au moyen d'un paramètre sur le logiciel d'acquisition.

Pour le modèle Varian® TrueBeam, le mode cine enregistre un film normalisé qui ne convient pas aux applications dosimétriques. Cependant un mode service permet d'extraire les images avec l'ensemble des trames individuelles soit en format DICOM qui est le format standard en physique médicale soit en format XIM [134]. Pour l'EPID d'Elekta®, une équipe de recherche a développé l'acquisition des trames, avec un taux de 2,5 trames par seconde, en interne [135].

Lorsque l'EPID au sillicium amorphe est sorti, les intérêts de recherches se sont concentrés sur la caractérisation de l'EPID acquis en mode intégré. Cela était bien adapté pour des traitements avec le bras de l'accélérateur statique. Cependant, avec l'arrivée du traitement VMAT, depuis 2009, le mode intégré n'est plus approprié car l'information du bras qui tourne autour du patient est perdue. Par conséquent le mode d'acquisition continu pour ce type de traitement est nécessaire.

La lecture des images ne se fait pas instantanément, la lecture est effectuée par portions à des intervalles de temps finis. Cela a son importance dans son utilisation dosimétrique, puisque les photodiodes conservent leurs informations de charge jusqu'à ce qu'elles soient lues. Par conséquent, l'influence dans la manière de lire le signal sur le détecteur plan a son importance. L'EPID Varian® est lu dans une configuration matricielle classique, puisque la lecture se fait ligne à ligne. Afin de gagner en efficacité de lecture, Varian® a décidé de faire une lecture de plusieurs lignes en parallèle avant de passer à la lecture du groupe de lignes suivantes. Environ 15 à 30 lignes forment chaque groupe de lecture. Une fois le nombre de lignes lues en parallèle fixé, il reste constant. Une équipe de recherche a décrit une méthode permettant d'ajuster le nombre de lignes lues entre les impulsions d'irradiation afin d'éviter la saturation des détecteurs [136]. Cette méthode a permis d'établir une relation entre le temps de lecture d'une ligne et une impulsion d'irradiation. En effet, le débit de dose n'étant pas forcément constant, il est intéréssant d'utiliser les impulsions d'irradiation comme références temporelles de lecture.

La lecture des images EPID Elekta® est plus complexe puisque le détecteur plan est découpé en 16 parties. La méthode de lecture est décrite en détail dans l'article [137]. Dans cet article, il est aussi démontré que la manière de lire le signal a un impact important pour les expositions de faibles UM.

3.3.3 Caractéristiques

Les EPIDs à base de a-Si sont aujourd'hui les plus rencontrés dans les services de radiothérapie.

Sur la Figure 3.2a, apparaît l'EPID de l'accélérateur Varian®. Les EPIDs a-Si-500/1000 [131, 138-146] utilisés durant l'étude était monté avec Exact-arm sur le modèle Clinac 23iX. Cet EPID possède une taille de $30 \times 40\ cm^2$ et comporte 384×512 pixels pour l'a-Si-500 et 768×1024 pixels pour la version améliorée a-Si-1000. Les faisceaux d'irradiation utilisés dans ces travaux font 6 MeV et 25 MeV. Le débit de dose durant l'irradiation était de 600 unités moniteur par minute et la vitesse d'acquisition des trames sur l'EPID était de 9,574 trames par seconde. La Distance Source-Détecteur (DSD) était de 150 cm pour la machine Varian®. Sur la Figure 3.2b, apparaît l'EPID de l'accélérateur Elekta®. L'EPID iViewGT [147, 148] était monté sur un Synergy de la marque Elekta®. Cet EPID possède une forme carré de $41 \times 41\ cm^2$ et est composé de 1024×1024 pixels. Dans cette étude, uniquement une énergie de 6 MeV a été utilisée. Le débit de dose durant l'irradiation était de 400 unités moniteur par minute. La DSD pour cette machine était de 160 cm.

Les caractéristiques souhaitées pour l'ensemble des dosimètres sont d'avoir une réponse linéaire en fonction de la dose absorbée reçue, une indépendance en fonction du débit de dose imposé par le traitement, une reproductibilité possible, une haute résolution spatiale, pas de temps mort à l'acquisition et que le signal soit obtenu en temps réel. Les chercheurs ont montré que plusieurs de ces caractéristiques étaient respectées par les EPIDs au sillicium amorphe. Pour l'a-Si-500/1000 de Varian® avec une acquisition en mode intégrée, de nombreux résultats sont disponibles [139]. La linéarité de la réponse de l'EPID en fonction de la dose absorbée a été établie. Toutefois, il arrive que pour les photons de faible énergie, on obtienne une sur-réponse de l'EPID [149-151]. Un petit temps mort a été identifié dans ces travaux et a été par la suite supprimé par le fabricant par l'intermédiaire d'une mise à niveau logicielle dans l'ordinateur d'acquisiion en passant de IAS2 à IAS3. Une équipe de recherche a démontré la pertinence du logiciel d'acquisition IAS3 pour l'objectif d'applications dosimétriques [152].

Concernant l'acquisition en mode ciné, une enquête de propriétés a été faite [153], il a été montré qu'il manquait une petite quantité constante de signal n'ayant pas d'influence significative sur le plan dosimétrique. En effet, seules les irradiations mesurées inférieures à 100 UM sont concernées, ce qui reste bien en dessous du seuil d'irradiation en VMAT typique. Cet effet a été appuyé et a montré que la cause était une acquisition d'image incomplète au tout début et à la toute fin de l'enregistrement. Concernant l'EPID iViewGT de Elekta® en mode intégré, de nombreuses études de performances dosimétriques ont aussi été documentées [147, 148].

La première équipe a noté une certaine non-linéarité de la réponse du signal en fonction de la dose absorbée. Cependant ils proposent une technique pour y remédier en ajoutant une plaque de cuivre de 2,5 mm d'épaisseur dans l'EPID permettant de couvrir la zone de *buildup*. De plus, la deuxième équipe a remarqué que la non linéarité était reproductible et pouvait être corrigée à l'aide d'une procédure d'étalonnage.

La non-reproductibilité à long terme sur les EPIDs Varian® a été démontrée comme étant inférieure à 1% sur tous les modèles confondus durant une période de trois ans [154]. Celle concernant les modèles EPIDs Elekta® a été démontrée comme étant inférieure à 0,5% sur une durée de deux ans [155]. Le pas de distances séparant les

pixels étant faible (environ $0,4 \times 0,4\ mm^2$), la résolution des images obtenues est élevée. Elle est même largement supérieure à la plupart des chambres d'ionisation ou encore, à celle du CT.

En outre, l'EPID s'avère être un équipement robuste face aux dommages pouvant être causés par les irradiations de haute énergies [156, 157]. Toutefois, si l'EPID vient à être utilisé régulièrement pour des applications dosimétriques de routine clinique (CQ ou DIV), il est nécessaire de mettre en oeuvre une procédure pour vérifier le bon fonctionnement et la qualité de réponse apportée par l'EPID. En effet, les dommages cumulatifs dûs aux rayonnements peuvent engranger une modification du comportement de l'EPID [158].

Les images EPIDs sont connues pour présenter des variations entre pixels dues à une différence de réponse intrinsèque des pixels individuels ou à des différences dans la réponse de l'électronique [159]. C'est pour cette raison que lorsqu'une trame de l'EPID est acquise, peut importe le mode d'acquisition, des corrections lui sont automatiquement appliquées pour chacun des pixels. En effet, un certain nombre de paramètres peuvent altérer sa qualité tels que les pixels défectueux, le bruit de fond, le courant de fuites des photodiodes, des offsets des électromètres et la différence de sensibilité entre les pixels. La trame de l'EPID acquise est régie par l'équation :

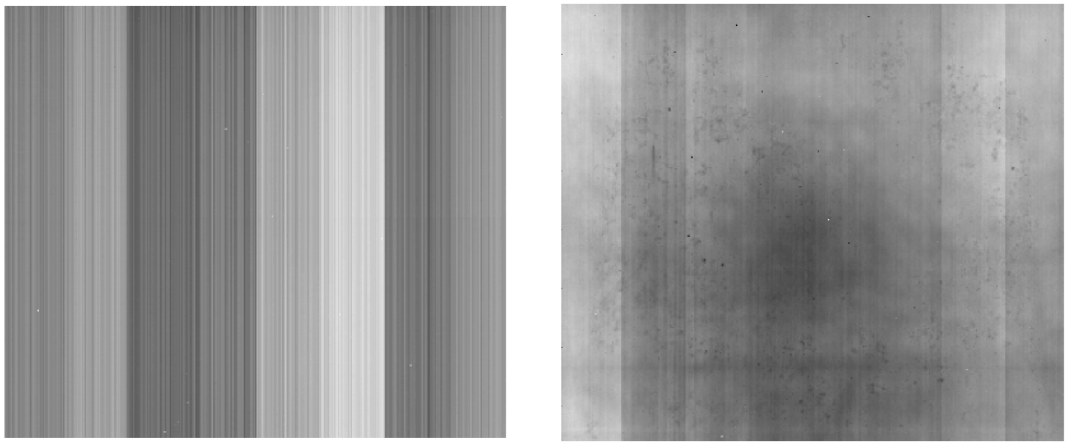
$$T(i,j) = \frac{T_{brute}(i,j) - DF(i,j)}{FF(i,j) - DF(i,j)}$$

avec $i \in 1, ..., N$, $j \in 1, ..., M$, N correspondant au nombre de lignes de la matrice de l'EPID et M au nombre de colonnes.

L'image appelée Dark-Field (DF), dont un exemple est visible en Figure 3.5a, correspond à une acquisition sans irradiation, à vide. Elle permet de mesurer le bruit de fond, le courant de fuites des différentes photodiodes et les offsets des électromètres. La seconde image, appelée Flood-Field (FF), visible en Figure 3.5b, correspond à une acquisition avec irradiation et munie d'un champ suffisamment large afin de couvrir l'intégralité de la surface du détecteur. Elle permet d'homogénéiser la réponse du détecteur.

Parmi les caractéristiques de l'EPID apparaissent des contraintes : ce n'est pas un détecteur équivalant eau et sa réponse n'est pas complètement uniforme sur toute sa surface de détection [142]. De plus, un effet de rémanence, souvent appelé *ghost effect* peut faire son apparition [147, 160]. C'est un effet combiné dû au gain de rémanence qui va modifier la sensitivité du pixel liée à la présence de charges piégées dans la matrice de photodiode. Il est aussi dû au *lag* de l'image lors de la lecture entraînant le dépôt de signal sur les trames suivantes. Il peut être nécessaire de supprimer la dernière trame dans le cas d'acquisition d'images en mode continu. Cet effet sera dépendant du temps d'irradiation et du débit de dose du traitement.

Parmi les contraintes apparaissent aussi le rétrodiffusé du bras de l'EPID. Il a été montré qu'il pouvait amener jusqu'à 6% de signal supplémentaire sur la dose absorbée maximum. Le rétrodiffusé est connu pour être asymétrique et dépendant de la taille du champ d'irradiation [161]. Pour finir, l'EPID possède un jeu mécanique. En effet, il est



(a) Mesure de la correction Dark-Field (b) Mesure de la correction Flood-Field

FIGURE 3.5 – Images de corrections apportées au signal brut de l'EPID

soumis à la force gravitationnelle pouvant entrainer un décalage de sa position idéale [162-165].

3.3.4 Réponse linéaire en dose absorbée de l'EPID

Lors d'une collaboration étroite avec la clinique Pasteur, on a effectué des mesures afin de vérifier la réponse de l'imageur portal EPID en fonction de la DDA planifiée pour les traitements RCMI sur un accélérateur Varian® [134].

Afin de mesurer la réponse, on a exposé l'EPID à différents nombres d'UMs, de 5, 10, 15, 20, 25, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500 et 600 UMs. La taille de champs a été de 10×10 cm^2 avec une énergie de 10 MeV et un débit de dose de 600 UM/min. Le signal recueilli au centre de l'EPID a été comparé à la profondeur de dose absorbée maximum calculé par le TPS dans les conditions de références avec la modélisation de la cuve d'eau présente physiquement lors de la mesure.

Les résultats obtenus ont permis de calculer une fonction de réponse en dose absorbée linéaire comme montré en Figure 3.6.

Les paramètres a et b sont respectivement le coefficient directeur de la droite et son ordonnée à l'origine. On remarque que la réponse en dose absorbée est linéaire par rapport à la mesure en niveau de gris EPID.

3.4 Dose absorbée de transit et de non transit

La dosimétrie de transit correspond à la mesure des photons résiduels, ayant traversé le volume irradié. La dosimétrie de non transit correspond à la mesure des rayons sans volume irradié (à vide). Elles ont pour objectif de vérifier que la dose absorbée délivrée correspond à la dose absorbée planifiée calculée par le TPS. Généralement, deux

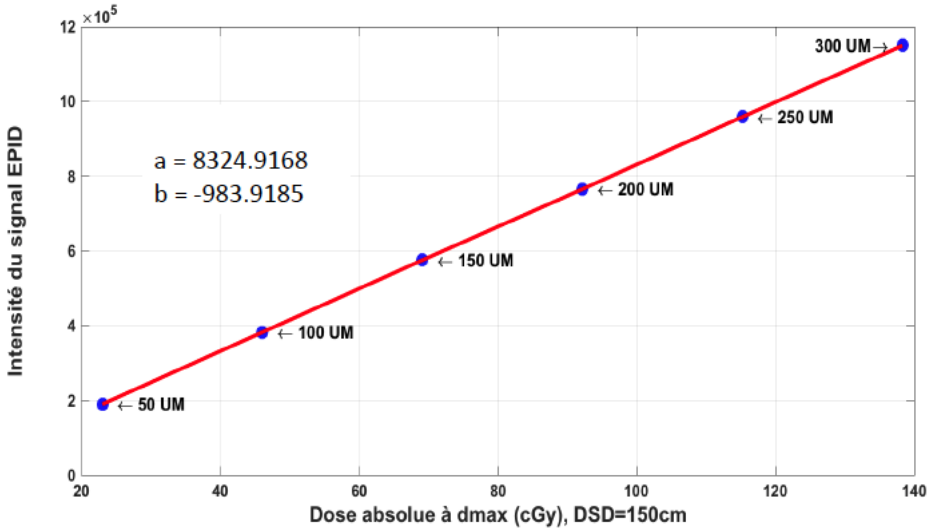


FIGURE 3.6 – Réponse en dose absorbée linéaire à partir d'une mesure EPID

approches de calcul sont souvent mentionnées dans la littérature : l'approche directe et l'approche indirecte.

3.4.1 Approche directe

Dans cette approche, les images de la dose portale sont prédites à la hauteur de l'EPID réél avant d'être comparées à la mesure corrigée. Le calcul prédictionnel se fait via plusieurs étapes :

— Le calcul de la fluence en sortie de la tête de l'accélérateur. Plusieurs possibilités pour l'obtenir, soit le fabricant fournit un modèle de la fluence via un fichier d'espace des phases pour le calcul Monte-Carlo, soit au préalable via la configuration du TPS.

— Le calcul de l'atténuation dans le volume irradié, soit à partir des simulations MC, soit à partir d'algorithmes de calcul analytiques de dose absorbée (Convolution/ Superposition, Pencil Beam, ...) comme introduit en 2.2.4.

— La modélisation de l'EPID, soit à partir des simulations MC, soit à partir de modèles analytiques fondés sur l'utilisation de kernels de dépôts de dose absorbée. Sont appelés kernels, des noyaux (carte de valeurs) ou fonctions mathématiques que l'on vient convoluer à l'image ou fonction de base.

Le calcul de dose absorbée au niveau de l'EPID à partir de simulations MC utilisant les données CT du patient a été exposé [131, 166]. Les auteurs ont commencé par simuler la fluence en sortie du patient, avant de la projeter au niveau de l'imageur portal et obtenir la prédiction.

Une autre approche [167] consiste à prédire en premier, la fluence primaire en sortie du patient. prenant compte de l'atténuation dans le patient à l'aide des distances

radiologiques. La distance radiologique corresppond à la somme pondérée des densités électroniques traversées et calculées à partir du CT, pondérée par la distance euclidienne de chaque voxel traversée. Par la suite, la correction du diffusé est calculée par le produit de convolution de la fluence primaire avec des kernels pencils beams générés par des simulations Monte-Carlo. Pour finir, le dépôt de dose dans le détecteur est calculé analytiquement via un produit de convolution entre la fluence totale et des kernels de dépôts de dose générés également par Monte-Carlo. Ils ont ainsi pu valider leur méthode pour des champs non modulés en 6 MeV et 23 MeV sur des fantômes de différentes épaisseurs dont certains comportaient des hétérogénéités de densité.

Une approche simple pour prédire les images portales a été développée [168]. L'EPID a été modélisé à sa position réelle dans le TPS comme étant un fantôme d'eau. Ensuite le calcul dosimétrique est effectué à partir du modèle de convolution/superposition présent dans leur TPS. Au niveau de l'EPID, une erreur de 4 % a été trouvée dans le champ d'irradiation.

Le principal inconvénient avec cette approche est qu'elle ne permet pas de reconstruire la DDA dans le patient que ce soit dans le volume cible ou dans les tissus sains. Elle peut cependant offrir la possibilité de diagnostiquer une faute et d'en identifier la cause durant le traitement. Afin de pallier cet inconvénient, plusieurs auteurs ont exploré des méthodes de reconstruction indirecte de la dose absorbée dans le patient en 2D et en 3D à partir des images de doses portales de transit.

3.4.2 Approche indirecte

Deux méthodes sont distinguées dans cette approche. L'une d'entre elles consiste à extraire la fluence primaire de l'EPID puis à calculer la dose absorbée à partir de cette dernière [169-173]. Le signal EPID mesuré correspond à la somme des contributions de la fluence primaire atteignant l'EPID et du rayonnement diffusé provenant du patient et de son environnement. Une matrice de déconvolution du rayonnement diffusé permet d'obtenir la fluence primaire. Cette fluence primaire obtenue est alors utilisée pour calculer la dose absorbée dans le patient avec des modèles de calculs conventionnels tels que l'algorithme Collapsed Cone Supersposition [174, 175]. Dans la littérature, plusieurs travaux de reconstruction de la dose absorbée en 3D ont été basés sur des algorithmes de calculs analytiques utilisant des produits de convolutions entre la fluence primaire et des kernels modélisant le dépôt d'énergie à l'intérieur du patient. Une équipe de recherche [176] a pour cela utilisé des kernels établis par [177] afin d'obtenir une DDA reconstruite en 3D dans un fantôme anthropomorphe. Une autre équipe [178] a étendu ces travaux en incluant l'imagerie du jour du patient. Pour cela, la fluence primaire est extraite de l'image du portal par estimation du diffusé avec une méthode de calcul itérative.

Pour la seconde méthode, l'image mesurée sur l'EPID est rétroprojetée soit ponctuellement pour une dosimétrie 0D, soit dans un plan du patient pour une dosimétrie 2D ou dans tout le volume pour une dosimétrie 3D. La rétroprojection se fait via une imagerie anatomique, soit avec le CT de référence, soit avec le CBCT pris le même

jour. La dDDA reconstruite peut alors être comparée à celle qui avait été planifiée via le TPS.

À partir de la valeur d'un pixel de l'EPID avec une correction des différents phénomènes physiques apparaissant sur l'EPID, il est possible d'obtenir un pixel avec une valeur de dose absorbée [151, 179-181]. Deux équipes [182, 183] ont établi une corrélation entre la valeur obtenue depuis l'EPID sur l'axe du faisceau et la valeur de dose absorbée obtenue à 5 cm de profondeur en utilisant des modèles empiriques. Deux autres équipes [184, 185] ont travaillé sur une méthode similaire pour reconstruire la DDA mais à la position isocentrale cette fois-ci.

D'autres auteurs utilisent une tout autre méthode qui consiste à utiliser la taransmission du patient en effectuant le rapport entre l'image reçue avec et sans patient. En agissant de la sorte, ils ont pu reconstruire la dose absorbée à mi-épaisseur radiologique en appliquant les différentes corrections analytiquement (atténuation, diffusés, divergence...) sans prendre en compte les hétérogénéités de densités [186, 187].

Cette méthode a été étendue au calcul de la dose absorbée dans un volume 3D par l'intermédiaire d'une reconstruction de chaque plan parallèle à l'EPID. Cela a pu être fait avec l'utilisation du contour externe du patient présent dans l'image DICOM du CT [188]. Pour finir, ils ont étendu leur méthode afin que le calcul de dose absorbée soit équivalent eau quelle que soit la nature du volume traité [189]. D'autres études ont montré un intérêt concernant la conversion de l'image EPID en image de dose absorbée équivalent eau [134, 190-193].

Cette approche comporte l'énorme avantage de pouvoir reconstruire une DDA dans le patient. Cependant, elle comporte aussi quelques inconvénients tels que le temps de calcul nécessaire lorsqu'il y a utilisation des méthodes MC. Concernant les approches analytiques, elles requierent une validation pour différentes situations cliniques car plusieurs approximations de calculs sont faites. De plus, toutes ces approches dépendent du modèle du LINAC et de l'EPID. C'est à dire, que chacun d'entre eux ont besoin d'être correctement modélisé. Ils requierent un certain nombre de paramètres et une calibration bien précise avant l'implémentation de l'algorithme.

3.4.3 Approche par réseau de neurones artificiels

On a vu précédemment, que l'EPID pouvait être utilisé à des fins dosimétriques. Dans la littérature, il commence à y avoir beaucoup de recherches le concernant Les recherches concernent l'analyse de ses caractéristiques, de ses propriétés physiques, de son comportement à long terme et les différents algorithmes modifiant le signal EPID pour d'obtenir une DDA. On a vu différentes techniques avec les approches directes et indirectes.

L'approche qui est utilisée dans ce manuscrit est une approche récente qui consiste à utiliser des algorithmes de réseau de neurones artificiels permettant de transformer un signal EPID en une DDA. La transformation de ce signal peut concerner la phase de prétraitement, permettant de contrôler la qualité du faisceau. Elle peut aussi concerner

la phase durant traitement appelée DIV permettant de contrôler que la dose absorbée délivrée par la machine correspond à celle planifiée. L'avantage d'une telle méthode est la possibilité de modéliser des phénomènes complexes à partir de données fournies. Un second avantage est de s'épargner des modèles si difficiles à produire pour les différentes marques d'accélérateur de particules et d'EPID. En effet, la modélisation à partir des données de chaque marque sera intrinsèque au RNA.

Quelques travaux ont déjà utilisé les RNA pour la vérification de traitements [194-196]. Les deux premières équipes ont utilisé une approche par RNA pour classifier les images EPIDs. Ils permettent de séparer les images EPIDs qui peuvent être considérées comme correctes, c'est-à-dire, conformes à ce qui a été planifié, ou défaillantes. Alors que la troisième équipe a développé une approche hybride, classifiant les pixels avec un faible ou un signal élevé avant de prédire leur valeurs. Leurs données provenaient d'un accélérateur de particules Varian®.

Ici, l'étude propose une nouvelle méthode utilisant les RNA permettant de reconstruire une DDA 2D en utilisant exclusivement l'EPID pour le CQ et en ajoutant des informations concernant le CT pour la dosimétrie *in vivo*. L'idée sous-jacente est de trouver un algorithme suffisamment générique afin de pouvoir l'utiliser pour différentes machines sans avoir à se préoccuper des modélisations ou calibrations à produire.

3.5 Détection d'erreurs à partir de l'EPID

L'EPID peut permettre la détection d'erreurs après analyse du signal obtenu. Les erreurs peuvent être classées en 3 types : les erreurs provenant de l'accélérateur de particules, les erreurs provenant du TPS et les erreurs provenant du patient.

— Les erreurs provenant du LINAC sont liées à des défauts mécaniques tels que la mauvaise rotation du bras, mauvaise position des mâchoires ou du MLC, mauvais débit de dose transmis, problème de chambre d'ionisation, présence ou absence d'accessoires de traitement, mauvaise position de la table, mauvaise calibration de l'EPID, défaillance de la mesure EPID et mauvaise position de l'EPID.

— Les erreurs provenant du TPS concernent les erreurs de calcul de dose absorbée, de modélisation des MLCs et de transfert de traitement d'un autre patient à l'accélérateur.

— Les erreurs provenant du patient sont plutôt d'ordre géométrique telles qu'un déplacement du patient durant le traitement, un mouvement de respiration, un changement anatomique par rapport au CT initial.

Les erreurs détectables énoncées ne sont pas exhaustives. Cependant, elles sont détectables via la mesure et l'information contenue dans l'EPID.

3.6 Conclusion

Ce chapitre présente un aperçu technique des EPIDs au sillicium amorphe dans un contexte dosimétrique. Les notions dosimétriques avec les différents points de mesure ont été données. Le sujet a par la suite été focalisé sur les systèmes EPIDs a-Si avec une description de leurs propriétés physiques, leur système d'acquisition et leurs caractéristiques dosimétriques. De plus, les différentes méthodes de reconstruction dosimétrique basées sur l'EPID ont succintement été montrées.

On a vu préalablement que les cliniques de radiothérapie modernes mettaient en oeuvre des traitements complexes tels que la RCMI, le VMAT ou des traitements stéréotaxiques avec des forts gradients de dose et contenant moins de fractions. Ces facteurs incitent fortement les chercheurs à continuer d'enquêter sur les EPIDs dans un objectif d'améliorer la vérification du traitement reçu par les patients.

Chapitre 4

Contrôle qualité avec les réseaux de neurones

4.1 Introduction

Le début de ce chapitre concerne l'étude des données utilisées pour ces travaux. Cette étape était nécessaire afin de tirer profit des informations qu'elles pouvaient contenir. En effet, la cohérence de l'information maintenue a son importance dans des modèles comme les réseaux de neurones. Oublier un paramètre peut fausser les résultats obtenus par l'application. Ensuite, l'application directe permettant la reconstruction de dose absorbée est détaillée. Le choix d'utiliser des modèles de FFNNs et CNNs a été fait. Le critère γ_{index} a permis d'évaluer la qualité des résultats obtenus. Ce critère est très utilisé en clinique par les physiciens médicaux car il permet de comparer pertinemment deux DDAs. Ensuite, les algorithmes ont été étendus pour son utilisation dans de nouvelles conditions. Ils ont été testés pour une autre énergie de traitement et pour une nouvelle marque d'accélérateur de particules. De plus, la capacité des algorithmes à détecter un défaut a été simulée. Pour finir, la comparaison entre les résultats provenant des deux modèles différents a été effectuée.

4.2 Matériels et méthodes

Durant ces travaux, le choix d'avancer étape par étape a été fait. L'idée était de commencer avec un modèle basique et d'étudier la faisabilité du projet avant de continuer avec des modèles plus complexes. Pour cette raison, il a été décidé de commencer les travaux avec les données de la RC. Cette étape a été conséquente car il a fallu prendre en compte un certain nombre de paramètres afin de fixer un modèle représentatif du système étudié. Dès lors, une fois l'étude terminée, les algorithmes ont été étendus pour des données de RCMI. Cela a demandé une étude particulière car un nouveau type de données devait être pris en considération dans les algorithmes. L'extension suivante concernait la prise en charge de données de traitement avec une énergie d'irradiation différente. En effet, l'utilisation d'une autre gamme d'énergie implique des changements comportementaux au niveau de l'EPID. De plus, le calcul de dose absorbée dans le patient ne suit pas la même dynamique et diffère en fonction de l'énergie. Pour finir, une

nouvelle extension a été proposée permettant de prendre des données provenant d'une autre marque d'accélérateur de particules généralisant l'utilisation des algorithmes.

4.2.1 Étude des données

Les réseaux de neurones étant un modèle d'apprentissage à base de données, il semblait nécessaire d'étudier plusieurs aspects les concernant afin d'obtenir un apprentissage optimal. Dans la littérature, on retrouve régulièrement stipulé le besoin d'un grand nombre de données pour obtenir un apprentissage suffisamment général pour que le modèle puisse prédire de nouvelles valeurs justes. La problématique rencontrée durant ces travaux a été la difficulté à se procurer une grande base de données pour différents aspects.

Le premier aspect concerne l'éthique dans le traitement des données médicales. En effet, différentes lois de protection des données personnelles sont mises en place, et ne permettent pas une récupération facile des données. Toutefois, une procédure d'anonimisation des données et leur utilisation dans le centre clinique a permis d'en obtenir.

Le deuxième point concerne le temps de récupération de ces dernières. En effet, des mesures sur l'imageur portal sont nécessaires et doivent être acquises en présence d'un physicien médical. Pour finir, une maîtrise du TPS a été nécessaire pour pouvoir exporter les données.

Pour toutes ces raisons, l'acquisition d'une grande base de données n'a pas pu être effectuée, il a été nécessaire de conditionner les algorithmes en fonction de la base de données en notre possession. Les données utilisées pour cette application, concernent d'une part les images acquises de l'EPID et d'autre part, les distributions de dose absorbée planifiées par le TPS.

4.2.1.1 Le format standard DICOM

Dans la majeure partie des cas, le format des données utilisées est le Digital Imaging and COmmunications in Medicine (DICOM). Ce dernier correspond à une norme mise en place pour uniformiser les données accessibles dans le domaine médical. Une extension DICOM_RT pour la radiothérapie a été faite. Le format DICOM_RT se divise en plusieurs catégories :

— Le RT_Dose sera utilisé pour les images contenant des DDAs.
— Le RT_Struct contient l'information des contours établis sur les images CT par les médecins.
— Le RT_Image est utilisé pour l'imagerie portal et les DRRs.
— Le RT_Plan contient toute l'information concernant le plan de traitement.

Il en existe d'autres, mais ceux présentés ont été utilisés durant ces travaux. Chacune de ces sous catégories de format possède leurs propres vignettes appelées *tags*. Les

vignettes sont repérables par des codes alpha-numériques qui ne changent pas.

4.2.1.2 Radiothérapie conformationnelle

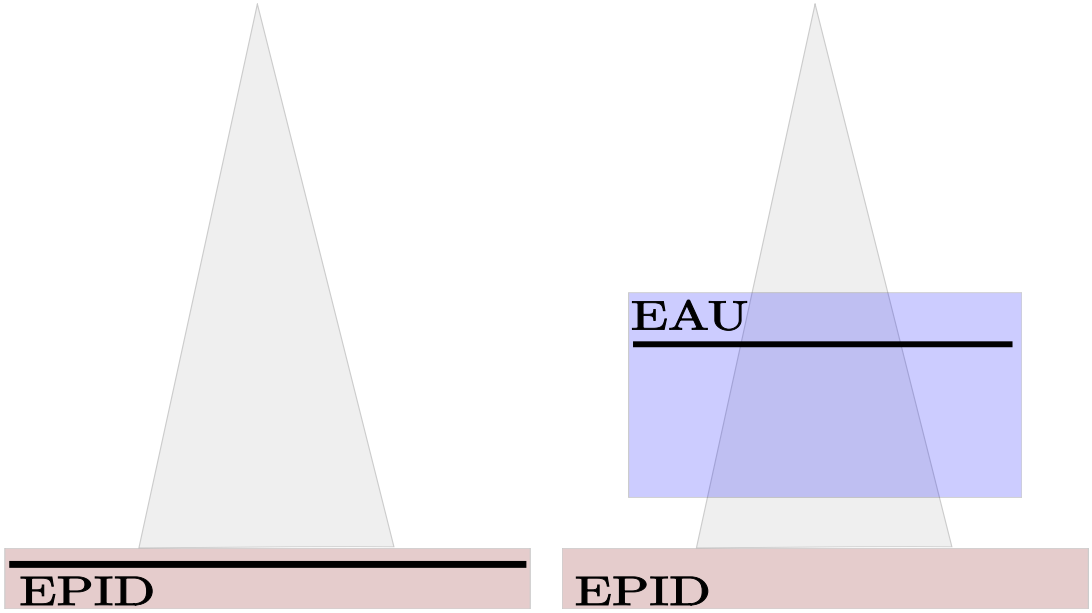
On a vu en 2.3.1, que la RC est l'une des techniques les plus simples encore utilisées. L'irradiation se fait de manière statique, signifiant qu'aucun mouvement de machine est fait durant le traitement (ni le collimateur secondaire, ni le bras de l'accélérateur). Les images EPIDs et les DDAs provenant du TPS ont un niveau de signal homogène au sein de la localisation à traiter et un fort gradient de dose à l'endroit où sont positionnées les lames. Ce type de traitement est encore utilisé pour l'irradiation du cancer du sein ou lorsqu'un traitement palliatif pour le cerveau entier est nécessaire. L'irradiation en RC se fait par l'intermédiaire de grands champs.

4.2.1.2.1 Machine Varian® La base de données récupérée pour la radiothérapie conformationnelle traitée avec l'accélérateur de particules Varian® possède huit ensembles de données EPIDs/TPSs. Cinq d'entre elles correspondent à des champs d'irradiation de cerveaux entiers et trois, à des champs de traitements localisés au sein. Les images récupérées sont au format DICOM avec toute l'information de l'image EPID contenue dans des RT_Image et toute l'information des DDAs contenue dans des RT_Dose.

Comme vu en 3.3.3, les images EPIDs obtenues sont composées de 768×1024 pixels. La possibilité, par l'intermédiaire d'un réglage, d'utiliser le mode de réduction de moitié de la résolution est proposée par Varian®. On a opté pour ce mode afin de pouvoir traiter de la même manière les images provenant des EPIDs a-Si 500/1000. On se retrouve donc, avec une résolution des images de 384×512 pixels. De plus, ce choix a été encouragé par le TPS. Les images de DDA proviennent du TPS Eclipse. Un fantôme d'eau cubique a été modélisé dans le logiciel Eclipse, l'isocentre a été placé à la profondeur correspondant à la distance où la dose absorbée est maximum, 1,5 cm comme on peut le voir sur la Figure 4.1b. L'image de distribution de dose absorbée est le plan orthogonal au faisceau positionné à l'isocentre.

Pour l'intégralité des champs d'irradiation concernant le contrôle qualité, la position du bras de l'accélérateur a été ramenée à 0 degré. Position dans laquelle le bras est vertical et la tête de l'accélérateur en position haute. L'exportation de l'image a été faite avec une résolution de 384×512 (résolution maximale provenant du TPS) et un redimensionnement du champ prenant en compte la projection conique du faisceau.

L'image EPID a été acquise en mode intégré et à une DSD de 150 cm. Elles étaient toutes corrigées du FF et du DF comme stipulé en 3.3.3. Durant le CQ, l'EPID est irradié directement sans objet placé sur la table, comme on peut le voir sur la Figure 4.1a. Un filtre seuil (*threshold*) à 10% a été utilisé sur l'image EPID permettant de désigner sur quels pixels seront calculées les prédictions de dose absorbée. En effet, au-delà de 20% de la valeur maximum du signal, on considère que l'information prise en compte est pertinente. Le masque est créé et appliqué à la DDA du TPS. Cela permet d'obtenir un nombre de pixels équivalents sur chaque échantillon de données.



(a) Représentation de la position des images EPIDs acquises

(b) Représentation de la position des images TPS exportées

FIGURE 4.1 – Représentation de la position des images acquises lors du contrôle qualité

4.2.1.2.2 Machine Elekta® La base de données récupérée pour la radiothérapie conformationnelle traitée avec l'accélérateur de particules Elekta® possède six ensembles de données EPIDs/TPSs. Quatre d'entre elles correspondent à des champs d'irradiation de cerveaux entiers et deux, à des champs de traitements localisés au sein. Le choix de prendre des ensembles de données ressemblant au premier cas a été fait. L'idée sous-jacente était de pouvoir comparer les résultats dans des cas similaires avec différentes machines. Les images EPIDs obtenues étaient au format DICOM.

Cependant, les images de distribution de dose absorbée étaient au format texte, contenant toute l'information nécessaire pour pouvoir composer l'image. Une en-tête était présente pour informer des méta-données. De plus, le pas et la position du premier pixel de chaque axe étaient donnés. Les résolutions n'étant pas identiques entre les images EPIDs et TPSs, on a été contraint de trouver un compromis permettant de perdre le moins d'information possible. Le compromis a été de redimensionner toutes les images en 512×512 pixels.

Un programme a été développé et a permis de mettre en forme les données. Mettre en forme signifie interpoler les valeurs, afin d'avoir des données EPIDs/TPSs de même taille, tout en prenant en considération la projection conique du faisceau. Les images de distribution de dose absorbée proviennent du TPS Pinnacle, les conditions d'exportation des données sont les mêmes que pour l'accélérateur Varian®, c'est à dire, qu'un fantôme d'eau a été modélisé, que le bras de l'accélérateur a été positionné à 0 degré et que le maximum de dose absorbée a été placé à la profondeur de l'isocentre. L'image EPID a été acquise en mode intégré et à une DSD de 160 cm. Les images ont été corrigées du FF et du DF. Le *threshold* des images a été fixé à 15%.

4.2.1.3 Radiothérapie conformationnelle à modulation d'intensité

On a vu en 2.3.2, que la RCMI était la forme évoluée de la RC. Evoluée, car elle permet de mieux se conformer à la tumeur grâce au système MLC qui est en mouvement durant l'irradiation. Les images de ce type de thérapie sont hétérogènes en terme de signal avec un fort gradient de dose à l'endroit où sont positionnées les lames. La RCMI est encore largement utilisée dans les centres cliniques. Elle permet de traiter un grand nombre de localisations.

4.2.1.3.1 Machine Varian® pour différentes énergies Les bases de données récupérées pour la radiothérapie conformationnelle à modulation d'intensité traitées en mode sliding window avec l'accélérateur de particules Varian® possèdent, respectivement, dix ensembles de données EPIDs/TPSs pour une énergie de 6 MeV et cinq pour une énergie de 25 MeV. Pour l'énergie de 6 MeV, quatre d'entre elles correspondent à des champs d'irradiation encéphaliques, trois à des champs de traitements localisés au sein et les trois autres sont des images ORL. Concernant l'énergie de 25 MeV, trois concernent des champs prostatiques et deux autres des champs encephaliques. Les images ont été récupérées dans les mêmes conditions que pour la RC. Elles sont au format DICOM, acquises en mode intégré, possèdent 384×512 pixels, ont été mesurées avec l'angle du bras de l'accélérateur positionné à 0 degré et ont une DSD de 150 cm. Elles ont été corrigées du FF et du DF. Le *threshold* des images a été fixé à 10%.

4.2.1.3.2 Machine Elekta® La base de données récupérée pour la radiothérapie conformationnelle à modulation d'intensité traitée en mode step and shot avec l'accélérateur de particules Elekta® possède 4 ensembles de données EPIDs/TPSs. Deux d'entre elles correspondent à des champs d'irradiation d'encéphale et les deux autres sont des images ORL. Les images ont été récupérées dans les mêmes conditions que pour la RC. Elles sont au format DICOM pour les images EPIDs, acquises en mode intégrée, possèdent 512×512 pixels, ont été mesurées avec l'angle du bras de l'accélérateur positionné à 0 degré et ont une DSD de 160 cm. Concernant les images de distribution de dose absorbée, elles ont été exportées au format texte. Le même traitement que la RC, pour la mise en forme commune des données EPIDs/TPSs a été faite. Elles ont été corrigées du FF et du DF. Le *threshold* des images a été fixé à 10%.

4.2.1.4 VMAT

On a vu en 2.3.3, que le VMAT était la forme évoluée de la RCMI. En effet, en plus d'avoir le MLC qui est en mouvement durant l'irradiation, elle a le bras de l'accélérateur qui tourne autour du patient apportant une hétérogénéité en terme de signal sur la 3ème dimension. L'arc-thérapie volumique modulée est de plus en plus utilisée dans les centres cliniques car elle permet, à la fois, un gain de temps de traitement considérable et un traitement plus précis. Toutefois, la vérification de cette technique devient plus compliquée car il est nécessaire de représenter la troisième dimension à partir du signal de l'EPID qui est en deux dimensions.

Dans le cadre du contrôle qualité du faisceau, puisque l'objet central est homogène

et correspond à de l'air, il sera possible de faire l'hypothèse suivante. Récupérer une image EPID intégrée en VMAT aura le même sens que son homologue EPID en RCMI car son mouvement de rotation du bras n'influera pas sur le signal reçu par l'EPID. Ceci revient à faire la mesure dans les mêmes conditions que pour la RCMI avec l'angle du bras de l'accélérateur positionné à 0 degré.

4.2.2 Prétraitement des données

La phase de prétraitement des données est une phase primordiale dans l'apprentissage automatique. Cette phase a été une des plus fastidieuse et chronophage durant cette thèse. Une des notions fondamentales pour modéliser correctement un système à partir de données avec ce type d'algorithme, est de trouver une cohérence entre les données d'entrées et les données de sorties fournies. Pour que la phase d'apprentissage soit efficace et fournisse un modèle caractérisant le système, il faut que les variables d'entrées et de sorties soient corrélées en fonction de l'intégralité des échantillons fournis durant cette phase. Il est indispensable que la phase d'apprentissage soit suffisamment générale pour obtenir les résultats attendus lors de la phase d'inférence. Pour cela, il convient d'avoir une base de données conséquente et représentative du système complet.

Une étude a été faite permettant de déterminer les données les plus pertinentes à fournir aux algorithmes. Plusieurs paramètres et normalisations ont été déterminés avec différents états. Un programme a permis de tester exhaustivement les combinaisons de ces différents paramètres afin d'en ressortir la meilleure solution. Ces différents paramètres sont les suivants :

— L'inversion du niveau de gris de l'EPID. Ce paramètre a été pris en compte car il existe une relation linéaire entre le niveau de gris et le niveau de dose absorbée. Cependant, durant un traitement, si un milieu à forte densité est traversé, le signal en niveau de gris EPID sera faible car une plus grande partie des photons aura intéragi avec la matière que si le milieu était pourvu d'une faible densité. Ce paramètre a été pertinent pour la dosimétrie *in-vivo* qui sera vue au Chapitre 5. Par contre, pour le CQ, il n'a pas d'impact car il n'y a pas de milieu traversé lorsque la mesure EPID est effectuée, comme montré en Figure 4.1a.

— Le *threshold* (seuil) qui est réglable entre 0 et 100%. Il permet de considérer les pixels ayant du signal pertinent pour la reconstruction de DDA. Considérer un seuil à 0% signifie considérer l'intégralité du signal et un seuil à 10% prend en compte les valeurs au-dessus de 10% de la valeur maximale du signal transmis en entrée.

— Paramètres permettant le respect de la proportionnalité entre la DDA TPS et la réponse en niveau de gris de l'EPID. Les paramètres correspondent, pour l'EPID, au nombre de trames « RTImageDescription » enregistrées pendant l'acquisition qu'il est nécessaire de multiplier par le signal. En effet, le signal moyen de toutes les trames acquises est mesuré. De plus, afin d'obtenir une réponse proportionnelle, pour la DDA TPS, il est nécessaire de multiplier un facteur de correction appelé « DoseGridScaling » par la DDA elle-même, comme vu en 3.3.2.

— Le diamètre de diffusion pris en charge dans le modèle, étudié en 4.2.3.1.

— Les traitements considérés en phase d'apprentissage et d'inférence.

— La normalisation, l'un des paramètres les plus importants. En effet, il est souvent considéré que la normalisation des données est intrinsèque aux réseaux de neurones, car les fonctions d'activation permettent d'uniformiser les valeurs de sortie des différents neurones. Néanmoins, lorsqu'un ensemble de données ne paraît pas être sur la même échelle de valeur, il est nécessaire de les normaliser afin de trouver une cohérence dans les données entrées/sorties transmises à l'algorithme.

Chaque traitement de patient étant différent, le nombre d'UMs n'est pas universel. Le signal en niveau de gris reçu sur l'EPID n'est donc pas uniforme et dépend du nombre d'UMs délivrées. C'est pour cela que différentes normalisations ont été testées dans le but de mettre à l'échelle l'intégralité des données. Trois fonctions de normalisation ont été programmées : la normalisation selon une loi exponentielle, la normalisation entre 0 et 1 en redirigeant la valeur maximale à 1 via la formule $z_i = \frac{x_i}{max(x)-min(x)}$ avec x le vecteur ou la matrice à considérer et la normalisation entre 0 et 1 en redirigeant la valeur maximale à 1 et la valeur minimale à 0 via la formule $z_i = \frac{x_i - min(x)}{max(x)-min(x)}$. De plus, pour chacune de ces fonctions de normalisation, ont été testées différentes combinaisons de normalisation sur les ensembles de données :

— La normalisation s'est faite sur l'ensemble des données, c'est à dire que la valeur maximale et/ou minimale de l'intégralité des données EPIDs/TPSs ont été prises. Les images ont été normalisées en fonction de ces dernières.

— La normalisation s'est faite par catégorie de données, c'est à dire que la valeur maximale et/ou minimale des données EPIDs et TPSs ont été prises séparément. Les images EPIDs ont été normalisées en fonction des extremums de l'intégralité des EPIDs et les images TPSs ont été normalisées en fonction des extremums de l'intégralité des images TPSs.

— La normalisation s'est faite pour chacune des données EPIDs et TPSs, c'est à dire que chacune des images est comprise entre les mêmes valeurs.

Cette question de normalisation a toute son importance. Au-delà, de sa faculté à mettre à l'échelle l'intégralité des données, c'est aussi en fonction d'elle que va dépendre la reconstruction de DDA. Seront-elles en valeurs absolues ou relatives ? Si une dénormalisation est appliquée alors des valeurs absolues pourront être considérées, sinon des valeurs relatives seront estimées. La dénormalisation au sein de notre application n'est pas si simple à appliquer. En effet, rappelons que la normalisation est appliquée à la fois sur les données d'entrées ainsi que les données de sorties. Rappelons que durant la phase d'inférence, aucune information provenant des données cibles ne doit être considérée, sinon les algorithmes seront influencés par ces derniers. On ne peut donc pas utiliser les coefficients de normalisation de la DDA provenant du TPS.

Cependant, toute l'information provenant de l'image EPID peut être utilisée. Or, nous savons que la réponse en dose absorbée est linéaire par rapport au signal EPID. Si le contraste de la carte de DDA arrive à être correctement calculé par les algorithmes, alors en appliquant les coefficients de linéarité, une reconstruction de DDA absolue est possible. N'ayant pas eu la possibilité d'obtenir ces coefficients pour chacun des

traitements étudiés, une alternative a permis de calculer des coefficients que l'on appellera de linéarité par analogie aux vrais coefficients de linéarité. Pour les calculer nous avons récupéré les coordonnées des 80 points avec les plus faibles et plus hautes valeurs de l'image EPID. Ensuite, le calcul d'une droite par la méthode des moindres carrées avec les 160 couples de valeurs EPIDs et TPSs de la phase d'apprentissage a été effectué. Les coefficients de pente et de position à l'origine obtenus ont été considérés comme les coefficients de linéarité. En considérant l'intégralité des couples EPIDs et TPSs de la phase d'apprentissage, on s'est aperçu que les coefficients obtenus étaient identiques d'un couple à un autre.

4.2.3 Modèle et architecture de réseaux de neurones

Le choix du modèle pour la reconstruction d'image s'est porté sur les réseaux de neurones. On a vu dans le Chapitre 3, que la recherche était active concernant l'utilisation de l'EPID à des fins dosimétriques. Plusieurs méthodes ont été ciblées telles que les méthodes analytiques ou les méthodes Monte-Carlo. On a pu observer que les méthodes MC apportaient des résultats satisfaisants, cependant, le temps de calcul est trop élevé pour une utilisation clinique. Les méthodes analytiques ont fait l'objet de recherches conséquentes amenant à des modèles de calculs proches des résultats attendus.

Cependant, avec ce type de méthodes, on a constaté que la modélisation pouvait être complexe et propre à chaque accélérateur de particules. Un grand nombre d'approximations et de calibrations doivent être faites pour que le temps de calcul reste correct. De plus, dans ce type de modèle, l'intégration des hétérogénéités tissulaires est très difficile. Pour toutes ces raisons, une nouvelle approche a été proposée utilisant les réseaux de neurones.

Le Chapitre 1, a montré que cette approche possède différents types de modèles lui permettant d'être efficace pour un grand nombre d'applications. Dans le cadre de ce travail, l'application concernée est la radiothérapie externe et plus spécifiquement le CQ effectué avant les séances de traitement du patient. Cette phase de contrôle qualité permet de vérifier si la machine délivre correctement la DDA planifiée préalablement par le TPS. Avant de parcourir les différents modèles choisis, il est nécessaire de déterminer dans quelle catégorie d'algorithme se trouve le problème posé.

Dans le Chapitre 1, ont été exposés les différents types et modes d'utilisation des algorithmes d'apprentissage automatique. Pour trouver l'algorithme qui se conformait au mieux à notre application, la description du problème a dû être fixée. Dans un premier temps, l'objectif était de développer des algorithmes d'apprentissage automatique capable de convertir, instantanément, un signal EPID en une DDA 2D pour le CQ. Cet objectif implique un apprentissage à partir de la mesure de l'EPID et demande de calculer un signal de DDA.

Pour commencer, il était nécessaire de savoir s'il existait une vérité à suivre pour le calcul du signal de DDA. Or, dans le Chapitre 2, ont été énumérées les différentes étapes d'un traitement. On a observé qu'il y avait une phase de planification de traitement qui

permettait de planifier la meilleure DDA pour le patient. Cette distribution correspond à celle qui doit être délivrée au patient et peut être considérée comme l'état de référence. De cette manière, on se situe dans le type d'apprentissage supervisé avec en données d'entrées des images EPIDs, et en données de sorties de la phase d'apprentissage des images de DDA sortant du TPS. Deux hypothèses doivent alors être faites :

— Les DDAs sortant du TPS doivent être considérées comme étant la vérité.

— Les données mesurées de l'EPID pour la phase d'apprentissage ne doivent pas avoir subi d'erreurs de traitements.

Avec ces deux hypothèses respectées, la DDA réellement délivrée par l'accélérateur peut être calculée lors de la phase d'inférence de l'apprentissage automatique. Or, on sait que la première hypothèse est fausse dans certaines conditions, par exemple dans le cas de calculs de dose absorbée dans des milieux hétérogènes en terme de densité électronique tels que les poumons [111]. Cependant, on considère pour le contrôle qualité, que cette hypothèse est vraie dans la mesure où le calcul de la dose absorbée s'effectue dans un milieu homogène.

Dans le chapitre 1, il a été montré que deux modalités existaient pour ce type d'apprentissage automatique : la classification et la régression.

On a remarqué en 3.5, qu'à partir de la DDA reconstruite, il sera possible de diagnostiquer et détecter des erreurs qui se sont produites durant les différentes sessions de traitements.

Si l'objectif de ces travaux consistait uniquement à détecter des erreurs spécifiques alors il serait possible de modéliser des cartes de résultats entrainant une classification séparant les traitements corrects des traitements incorrects.

Cependant l'objectif est plus large puisqu'il s'agit de reconstruire une DDA. Cette dernière pourra être comparée à celle qui a été planifiée. Si des discordances sont présentes, alors il faudra diagnostiquer la cause à partir du calcul effectué.

Il est important de noter que pour tous les cas traités, l'image EPID qui a été utilisée pendant la phase d'inférence ne faisait pas partie de la base des données d'apprentissage. De plus, pour l'ensemble des données utilisées durant la phase d'apprentissage, le ratio était de 70% de données réservées à l'entrainement, 15% réservées au test et 15% à la validation,

L'objectif étant de reconstruire une DDA, la catégorie d'apprentissage choisie est la régression. En effet, les cibles devant être reconstruites sont des valeurs réelles.

Pour commencer l'étude, il a été choisi d'utiliser un modèle simple qui concerne les FFNNs. Par la suite, un modèle un peu plus complexe considérant un CNN a été implémenté.

4.2.3.1 Réseaux de neurones feed-forward

Les algorithmes déployés dans cette partie ont été implémentés avec Matlab® (MATLAB R2020b, MathWorks) en utilisant la toolbox Deep Learning version 14.1. Ce choix a été fait pour deux raisons : une raison historique car les premiers algorithmes avaient été développés sur Matlab® et une deuxième raison, plutôt pratique, car tous les frameworks cités en 1.4.9 ont été rendus accessibles au public au début de ma deuxième année de thèse.

Le réseau de neurones classique utilisé dans cette partie, est un modèle d'apprentissage supervisé. Il a été expliqué en 1.4.3, que pour ce type de modèle, les données d'entrées et de sorties sont à fournir durant la phase d'apprentissage. Cependant, uniquement les données d'entrées sont nécessaires durant la phase d'inférence. L'algorithme calculera alors le résultat à partir des données d'entrées et de ce qu'il aura appris durant la phase d'apprentissage.

Comme montré sur la Figure 4.2, on dispose de la mesure de l'EPID permettant de transmettre l'information des rayons ayant traversé le patient. Cette donnée a été utilisée comme donnée d'entrée des algorithmes. La DDA du TPS pour chaque traitement de patient correspond aux données de sorties de la phase d'apprentissage.

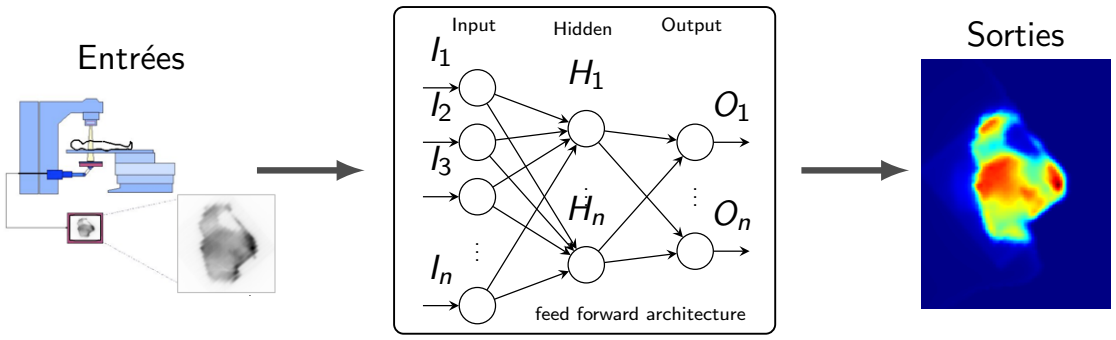


FIGURE 4.2 – Schéma explicatif des données d'entrées/sorties pour les réseaux de neurones classiques

L'objectif étant de calculer la DDA délivrée durant la phase d'inférence, il a fallu utiliser un grand nombre de données. En effet, dans le Chapitre 1, on a montré que plus la base de données était grande et justifiée, plus la probabilité d'obtenir un modèle correct était élevée. De plus, il a été expliqué en 4.2.1, la difficulté de récupérer une grande base de données.

Pour cette raison, les données transmises à l'algorithme ont été transposées. Habituellement, notamment pour la classification, on désigne un échantillon (correspondant à un ensemble d'entrées) comme étant une image complète. L'intégralité des échantillons correspond à l'ensemble complet des données. Si on avait procédé de la sorte, on aurait eu des dizaines d'échantillons pour des centaines de milliers d'entrées (une image étant

composée de 384 × 512 pixels).

Afin de mettre en place un modèle correct représentant de manière plus efficace le système, la considération de chaque pixel en tant qu'échantillon a été privilégiée plutôt que de considérer les images EPIDs et TPSs entières. En procédant de la sorte, on transpose le problème en se retrouvant avec des centaines de milliers d'échantillons pour des dizaines d'entrées. Cela permet de faire un apprentissage avec un grand nombre d'échantillons, donc efficace malgré un nombre d'images EPIDs et TPSs limité. Il reste important de considérer un panel d'images différentes afin de généraliser le modèle à de nouvelles images.

Le système est modélisable de cette manière car chaque pixel d'une image EPID correspond à un signal en niveau de gris qui est physiquement lié à chaque valeur de pixel de dose absorbée du TPS. Outre la relation physique directe entre les pixels de chaque image EPID et les DDAs TPS, les informations contenues dans les pixels voisins ont été introduites dans le modèle.

Lors de l'étude des paramètres à utiliser, on s'est aperçu que le nombre de variables d'entrées avait son importance. C'est pourquoi, faisant face à un système complexe et non linéaire, il a fallu déterminer quel était l'ensemble de données d'entrées permettant d'arriver au meilleur apprentissage. En effet, essayer de faire corréler une seule variable d'entrée avec une seule variable de sortie n'était pas approprié pour modéliser l'application souhaitée.

L'insertion des pixels voisins a permis de modéliser intrinsèquement le rayonnement diffusé du patient (dans le cas du CQ, il est équivalent à une cuve d'eau - milieu homogène). Toutes ces informations, ainsi que la localisation spatiale du pixel central ont été définies comme données d'entrées. Ce dernier n'a que très peu d'influence sur la modélisation car chaque champ d'irradiation est différent.

L'intégralité des données d'apprentissage (EPIDs et TPSs) ont été mises à l'échelle afin de rendre le modèle plus pertinent. Cette mise à l'échelle des données a été un pas essentiel de ces travaux puisqu'elle a permis de faire cohabiter différentes images EPIDs et TPSs au sein du même modèle. On a vu dans le Chapitre 3 que la réponse en terme de niveau de signal de l'EPID était dépendant de la quantité d'irradiation fournie durant le traitement.

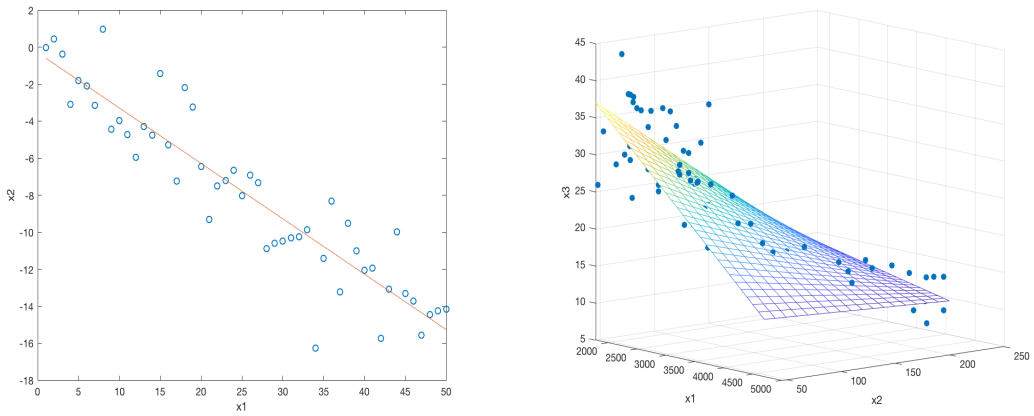
Cependant, on a expliqué que la réponse en dose absorbée était linéaire car la modélisation du fantôme d'eau dans le TPSs était conforme à celui présent sur la table de traitement. Or, on a vu que lors de la phase de contrôle qualité, la mesure se faisait sans fantôme contrairement au calcul de DDA dans le TPS qui se faisait avec la modélisation d'un fantôme d'eau. Cela ne remet pas en cause la linéarité de la réponse en dose absorbée par rapport à l'EPID, cependant, il est primordial de mettre à l'échelle les données pour que les intervalles de signal propres à chaque traitement soient fusionnés et cohérents. En effectuant cette étape, toutes les images étaient cohérentes entre elles.

Afin de ramener toutes les données sur la même échelle, la normalisation redirigeant

la valeur maximale à 1 et la valeur minimale à 0 a été effectuée pour chacune des images EPIDs et TPSs.

Revenons maintenant sur la méthode utilisée et ayant apportée les meilleurs résultats pour l'application. Malgré le faible impact qu'ont certains paramètres choisis, on a voulu dans un grand nombre de cas, anticiper l'utilisation vers la DIV. On a par exemple, utilisé l'inversion du signal EPID pour l'intégralité des travaux présentés. Le seuil (threshold) appliqué est propre à chacun des traitements. Leurs valeurs ont été exposées en 4.2.1. Les paramètres correspondant au nombre de trames de l'EPID et « DoseGridScaling » ont été appliqués afin d'obtenir la proportionnalité entre la DDA du TPS et la réponse en niveau de gris de l'EPID. Le dernier paramètre concerne le nombre de pixels voisins pris en charge dans le modèle. Ce paramètre a son importance puisque selon le nombre de pixels pris en considération, les résultats varient. Ce paramètre est directement en relation avec le phénomène de diffusion que l'on cherche à modéliser.

Différents phénomènes doivent être pris en compte. Le premier concerne la dimensionnalité mathématique des entrées. Comme on peut le voir sur la Figure 4.3a les données sont représentées sur un espace à deux dimensions alors que sur la Figure 4.3b, ces mêmes données sont représentées sur trois dimensions. On remarque que l'espace à trois dimensions permet la représentation de systèmes plus complexes qui ne seraient pas modélisables avec un espace à deux dimensions. On imagine que plus on augmente la dimensionnalité, plus il sera possible de modéliser des systèmes complexes.



(a) Exemple d'un problème à base de données à deux dimensions

(b) Exemple d'un problème à base de données à trois dimensions

FIGURE 4.3 – Exemple de régression linéaire

Cependant, on observe que si un système est modélisable sur un espace tri-dimensionnel, il ne l'est pas forcément sur un espace bi-dimensionnel. S'il n'est pas modélisable car l'espace est trop petit, l'algorithme sera confronté à un sous-apprentissage et le modèle ne s'adaptera pas aux nouvelles données. Au contraire, si l'espace considéré est trop grand, l'algorithme sera confronté à un sur-apprentissage et le modèle ne s'adaptera pas non plus aux nouvelles données. Ce problème de sur-apprentissage arrive fréquemment car en présence de bruit dans les données, une mauvaise modélisation y est encore

plus simple dans un espace de trop grande dimension. C'est pour cette raison qu'il est important de trouver le bon nombre et les entrées qui permettront de faire apprendre à l'algorithme un modèle représentatif du système.

Le deuxième phénomène qui intervient est physique. Il concerne la diffusion des rayonnements. L'énergie des rayonnements n'est pas juste déposée localement mais se propage aussi de manière diffuse autour de sa cible principale. Ce phénomène de diffusion est tri-dimensionnel et intervient de manière plus ou moins importante en fonction de la densité du matériel traversé, de la distance parcourue, de l'énergie utilisée et bien d'autres paramètres. Si bien, qu'en complément de l'énergie qui est déposée localement lorsque l'on considère des coordonnées spatiales, va intervenir de la diffusion de toutes les localisations spatiales voisines jusqu'à une certaine distance.

C'est pour cette raison, que le maximum de dose absorbée intervient à une certaine profondeur, c'est lorsque l'équilibre électronique est établi. C'est à dire que toutes les contributions du phénomène de diffusion interviennent en 3D. Comme on peut le voir sur la Figure 4.4, si on se place à une certaine coordonnée spatiale, une énergie sera déposée localement par le rayonnement primaire, le rayonnement secondaire provenant des coordonnées spatiales voisines y ajouteront leur énergie, puis le rayonnement secondaire de la coordonnée spatiale considérée ira déposer son énergie sur les coordonnées spatiales voisines. On dit alors qu'il y a équilibre électronique, lorsqu'il y a autant de participation provenant de l'extérieur que d'intervention énergétique vers l'extérieur.

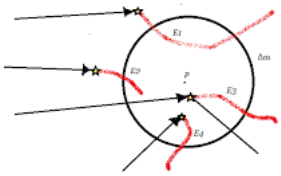


FIGURE 4.4 – Équilibre électronique

Ce phénomène est fortement impliqué lorsqu'un fantôme ou un patient est présent sur la table. Un fantôme ou un patient étant un volume 3D, l'équilibre électronique est atteint à une certaine profondeur. Cependant, les mesures sont faites sur l'EPID qui est un imageur plan. Cet imageur plan subira des phénomènes de diffusions latérales mais peu interviendront en profondeur. L'équilibre électronique ne sera pas atteint.

Cependant, il est nécessaire de produire un modèle capable de prendre en considération ce phénomène de diffusion. Pour cela, on a proposé de prendre en supplément du pixel concerné, ses pixels voisins comme étant des entrées supplémentaires. Considérons la taille d'un pixel de l'EPID, qui est de 0,768 mm. Construisons ce que l'on appellera la matrice de diffusion qui a pour taille $2*n+1 \times 2*n+1$ pixels avec $n \in \mathbb{N}$ le paramètre fixé dans les algorithmes. L'apprentissage a été fait pour tout n allant de 0 à 20 pour chacun des traitements. Il est important de remarquer que la valeur de n optimale varie en fonction de l'énergie de traitement et non en fonction du type de traitement exercé. En effet, pour les traitements en 6 MeV, le n optimal trouvé est de 5 alors que pour un traitement en 25 MeV le n optimal est de 7. Un n de 5 implique un diamètre de

diffusion de $(2 * 5 + 1) * 0,768 * \sqrt{2}$, donc un diamètre de diffusion d'environ 1,2 cm, ce qui reste cohérent à cette énergie. Un n de 7 implique un diamètre de diffusion de $(2 * 7 + 1) * 0,768 * \sqrt{2}$, donc un diamètre de diffusion d'environ 1,6 cm, ce qui reste dans la même cohérence que précédemment.

On sait que dans des conditions de mesures de références, la réponse en dose absorbée est linéaire par rapport au signal de l'EPID. Considérons que les hypothèses faites précédemment sont respectées, c'est à dire que le TPS planifie des DDAs correctes et que les traitements considérés durant l'apprentissage n'ont pas subi d'erreurs. On sait aussi que si l'on prend en compte toute la physique sous-jacente, il existe une corrélation entre les images EPIDs et les DDAs du TPS fournies pour l'apprentissage.

Considérons, que toute l'information pertinente du CQ soit concentrée dans la mesure, alors il sera possible de reconstruire la dose absorbée délivrée. C'est ce que fait le modèle utilisé dans ce manuscrit. Durant l'apprentissage, l'algorithme minimise l'erreur entre la cible (dose absorbée planifiée) et le calcul à partir de l'intégralité des images transmises en entrée correspondant à la mesure. Ainsi, pour chaque nouvel échantillon de données le calcul est itérativement réitéré.

Pour cette raison, si les pixels voisins ont une influence sur le pixel considéré en sortie, leurs influences seront propagées sur les poids des pixels respectifs en entrée. C'est pourquoi, il est nécessaire de prendre en considération autant de pixels voisins pour un apprentissage approfondi de la diffusion. Les algorithmes sont alors capables de créer un pattern général qui peut calculer des nouvelles DDAs cohérentes à partir de nouveaux jeux de données EPIDs.

Les différents paramètres de prétraitement des données ont été passés en revue. Ils ont montré leur importance au sein des algorithmes de calcul. Ce sont maintenant les hyper-paramètres des différents modèles qui vont être abordés.

L'utilisation de l'architecture la plus basique des réseaux de neurones a été étudiée et privilégiée en premier afin de se décharger de la complexité du modèle. Ce modèle correspond à l'architecture des FFNNs aussi communément appelée perceptron multi-couches. On a vu dans le Chapitre 1 que différents hypers-paramètres étaient à déterminer. Ainsi, il a été nécessaire de tester un certain nombre de combinaisons pour choisir les plus pertinentes.

Cette étude a permis de privilégier l'utilisation d'une couche de neurones d'entrées, une couche de neurones cachés et une couche de neurones de sortie. Le nombre de neurones d'entrées est dépendant de l'énergie de traitement. La couche d'entrée est composée respectivement de 124 et de 227 neurones, dans le cadre d'un traitement à 6 MeV puis 25 MeV. Dans les 124 et 227 neurones d'entrées, 1 neurone correspond au pixel central, 121 et 224 aux pixels voisins puis 2 aux coordonnées spatiales de l'EPID. Le nombre de neurones de la couche cachée est de 186 neurones pour une énergie de 6 MeV et 340 neurones pour une énergie de traitement de 25 MeV. La couche de sortie est composée d'un unique neurone de sortie correspondant au pixel de dose absorbée. Le nombre de neurones de la couche cachée a été fixé expérimentalement, une variable de 0 à 20 a été incrémentée par pas de 0,1. Cette variable a été multipliée au nombre

d'entrées afin de fixer le nombre de neurones cachés. Le résultat optimal a été obtenu pour l'intégralité des traitements lorsque la variable avait une valeur de 1,5. On a choisi par la suite de fixer le nombre de neurones cachés à 1,5 fois le nombre de neurones d'entrées.

L'algorithme d'optimisation utilisé durant la phase d'apprentissage est la méthode de gradient conjugué « Scaled Conjuguate Gradient ». Cet algorithme a été mis en opposition avec les autres, et les deux méthodes qui ont donné les meilleurs résultats sont le gradient conjugué et la méthode Levenberg-Marquardt. Cependant, cette dernière donnait des résultats corrects mais un temps de calcul entre 4 et 5 fois plus élevé. Les fonctions d'activation donnant le meilleur apprentissage concernent d'une part la fonction sigmoïde pour la couche cachée, et d'autre part la fonction linéaire pour la couche de sortie. L'initialisation de chacun des poids du modèle a été faite de manière aléatoire. La fonction coût devant être optimisée était la fonction MSE.

4.2.3.2 Réseaux de neurones convolutionnels

On a étudié dans la partie précédente les paramètres et hyper-paramètres à appliquer au modèle de réseau de neurones classique pour avoir un apprentissage efficace et représentatif du système. Dans cette partie, nous allons présenter l'étude d'un nouveau modèle pour la même application (la phase de contrôle qualité). Ce modèle concerne un CNN. Comme montré sur la Figure 4.5, on remarque que les données en entrée et en sortie du modèle sont identiques à celles transmises au réseau de neurones classique. Seules les coordonnées spatiales ont été retirées des données en entrée. De plus, les paramètres choisis pour le prétraitement sont identiques à ceux fixés précédemment.

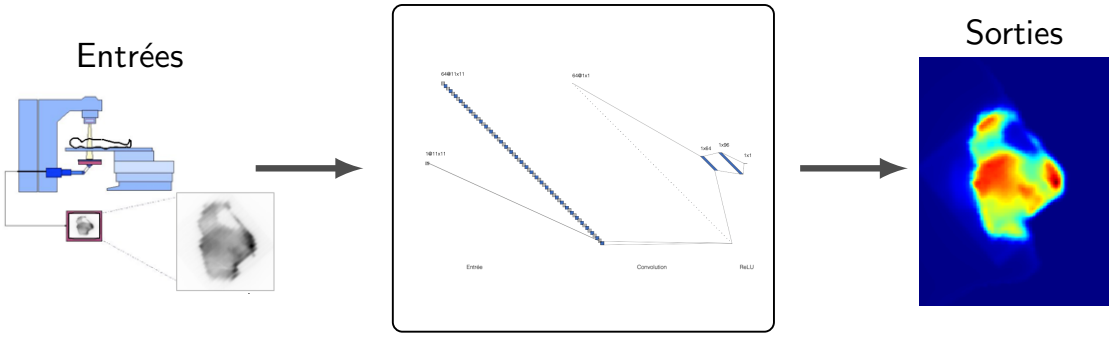


FIGURE 4.5 – Schéma explicatif des données d'entrées/sorties pour les CNNs

Cependant, l'architecture du modèle a été mis à jour. Sur la Figure 4.6, est représentée l'architecture du CNN. Cette architecture est représentée pour un unique échantillon de données d'entrées et de sorties. Un échantillon de données d'entrées est composé du pixel central considéré accompagné de 121 pixels voisins pour un traitement de 6 MeV, et accompagné de 224 pixels voisins pour un traitement de 25 MeV. Chaque échantillon est transmis à l'algorithme sous forme matriciel de respectivement 11×11 et 15×15

pixels.

La première couche concerne une couche de convolution composée de 64 cartes de caractéristiques obtenues à partir de filtres 11×11 et 15×15 pixels selon l'énergie de traitement considéré. Un *stride* de 1 sans padding a été implémenté. Les filtres à convolution n'avaient pas de taux de dilatation. La deuxième couche est une couche d'activation ReLU. La fonction mathématique ReLU est appliquée terme à terme à la matrice sortant de la précédente couche de convolution.

Les couches suivantes correspondent aux couches entièrement connectées. Ces couches sont comparables à un réseau de neurones classique. On remarque que la première couche entièrement connectée possède 64 neurones correspondant aux 64 caractéristiques déterminées par la partie convolutionnelle du réseau. Par analogie, avec les réseaux de neurones classiques, la deuxième couche entièrement connectée peut être vue comme étant une couche cachée. Cette couche possède 96 neurones. Pour finir, on remarque la couche de sortie possédant un pixel correspondant soit à une valeur de dose absorbée normalisée sortant du TPS pour la phase d'apprentissage, soit à la valeur calculée par les algorithmes durant la phase d'inférence.

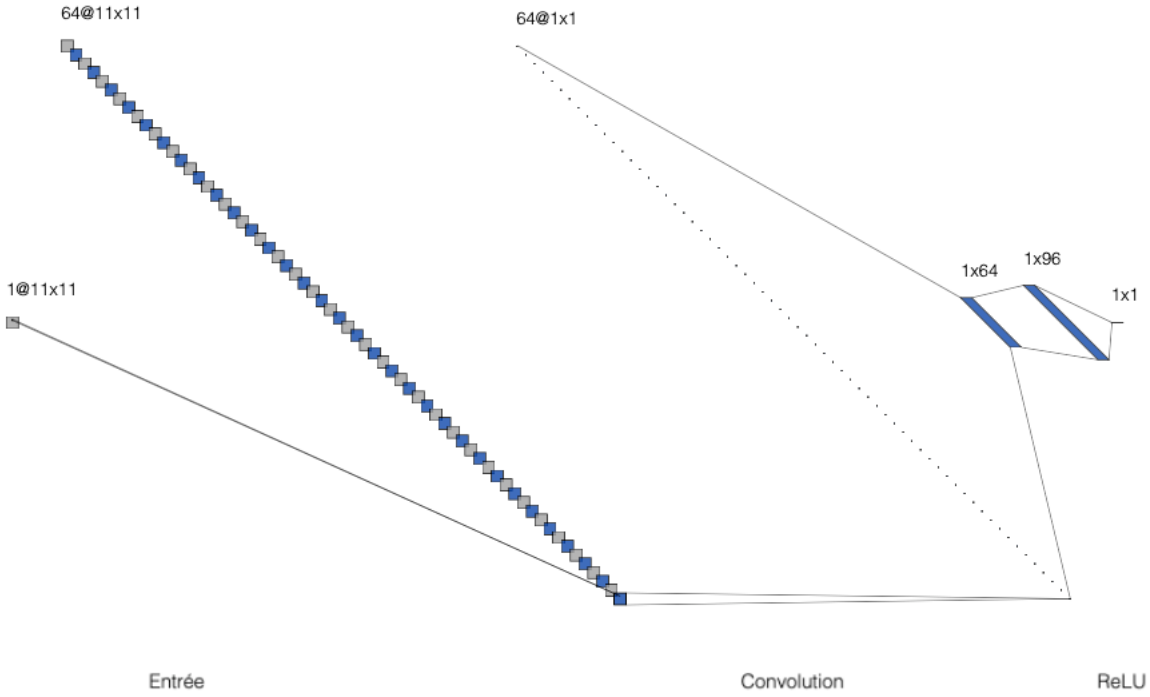


FIGURE 4.6 – Architecture du CNN

L'algorithme d'optimisation utilisé durant la phase d'apprentissage est la méthode Adam. L'initialisation de chacun des poids du modèle a été faite de manière aléatoire. La fonction coût devant être optimisée était la fonction RMSE.

4.2.4 Utilisation du critère clinique gamma

Le critère γ_{index} est un des critères permettant de comparer deux distributions de signaux. Il a été choisi car les physiciens médicaux l'utilisent comme étant le plus pertinent en routine clinique. Il permet la comparaison de signaux 1D, 2D et 3D. Dans le cadre de la routine clinique, il évalue la qualité des calculs de DDA effectué par les différents algorithmes [197-199]. Le critère γ_{index} a été développé juste avant le début des années 2000 [200, 201]. Il prend en considération deux paramètres qui sont la différence de dose absorbée en pourcentage et la distance euclidienne (représentation spatiale) appelée Distance To Agreement (DTA) entre deux points. La DTA correspond à la distance maximale acceptée séparant le point de dose absorbée de référence à celui calculé. Cette DTA est imposée par l'utilisateur.

Pour des distances supérieures à cette DTA, le critère ne sera pas satisfait. Il existe deux façons de calculer le γ_{index} : le γ_{index} local et le γ_{index} global. La différence se situe dans le calcul du pourcentage de dose absorbée maximum. Le γ_{index} global considère la valeur maximum sur l'intégralité des points de référence alors que le γ_{index} considère la valeur maximum considérant les points de référence à l'intérieur de la circonférence de la DTA. Cela implique que le γ_{index} local est plus restrictif que le γ_{index} global. Dans le cas simple d'une comparaison en 1D, une ellipse d'acceptabilité est définie autour de chaque point de référence. Le point à évaluer remplit les critères d'acceptabilité s'il est situé dans l'ellipse. Cette ellipse est régie par l'équation :

$$\gamma = \left(\frac{(D_m - D_c)^2}{D_{max}^2(\%)} + \frac{(r_m - r_c)^2}{DTA^2} \right)$$

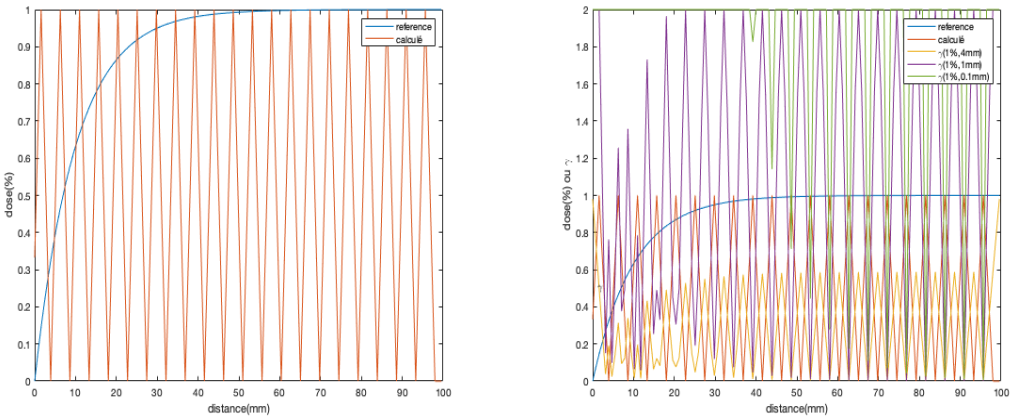
avec D_m la dose absorbée au point de référence (mesure), D_c la dose absorbée au point calculé, r_m les coordonnées du point de référence et r_c les coordonnées du point calculé.

Pour chacun des points considérés, uniquement la valeur minimale des γ obtenus est gardée puis placée dans la carte du γ_{index}. Pour chacune des valeurs retenues, si elle est inférieure ou égale à 1, on considère que le critère est respecté, autrement dit, que la comparaison faite entre le point mesuré et le point calculé est acceptée dans le domaine de tolérance fixé. Dans le cas contraire, si sa valeur est strictement supérieure à 1, alors le critère n'est pas satisfait.

Le résultat final du γ_{index} est souvent donné en pourcentage du nombre de points de la carte respectant la tolérance fixée.

On remarque rapidement que plus les valeurs données à D_{max} et DTA sont élevées, plus l'ellipse est grande. Cela implique que le γ_{index} sera plus élevé. Les physiciens médicaux en routine clinique fixent souvent la tolérance à $\gamma_{index}(3\%,\ 3\ mm)$ ou quelques fois, $\gamma_{index}(2\%,\ 2\ mm)$. Généralement, pour la vérification du contrôle qualité, si $\gamma_{index}(3\%,\ 3\ mm)$ est supérieur à 95% alors les physiciens considèrent que l'accélérateur de particules a transmis le traitement attendu.

Toutefois il est important de vérifier les conditions d'utilisation de ce critère. En effet, il est pertinent lorsque les signaux ne sont pas bruités. Dès lors où un des deux signaux de comparaison possède des forts gradients, le γ_{index} ne sera pas consistant. On peut voir sur la Figure 4.7, un exemple typique de ce qui vient d'être énoncé. Sur la Figure 4.7a se trouve en bleu les données de références et en orange les données calculées. Ces données sont fournies avec un pas d'échantillon de 0,768 mm, en gardant la même résolution que les images EPIDs. On remarque rapidement que ces données à comparer sont fondamentalement différentes, cependant on constate sur la Figure 4.7b que le γ_{index} est élevé malgré des courbes distinctes. Le $\gamma_{index}(3\%,\ 3\ mm)$ est à 98% montrant qu'il est nécessaire d'analyser les données avant de conclure la pertinence du γ_{index} obtenu.



(a) Données typiques pour le γ_{index} (b) Résultats du γ_{index}

FIGURE 4.7 – Exposition du γ_{index} dans un cas typique

Pour chacun des résultats présentés par la suite, le γ_{index} a uniquement été calculé sur la région d'intérêt. Les valeurs situées en dessous du threshold n'ont pas été prises en compte pour ne pas fausser la valeur du γ_{index}.

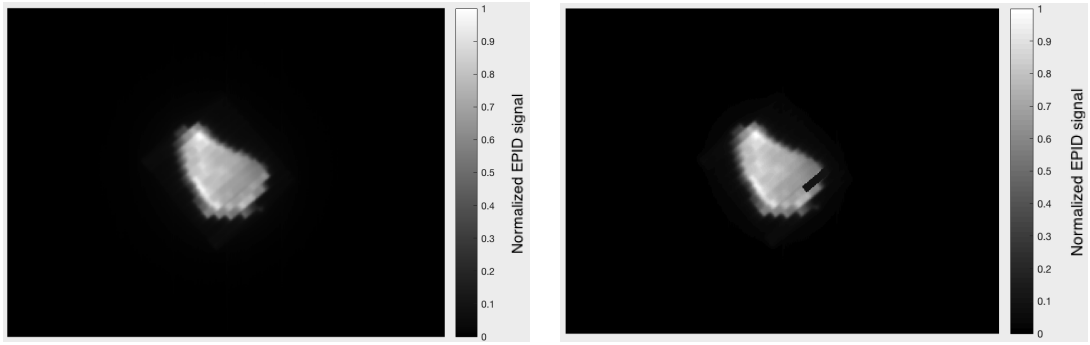
4.2.5 Détection d'erreur machine

Les différents modèles ayant pour objectif de calculer une DDA délivrée ont été exposés précédemment. Afin de vérifier la cohérence des algorithmes à reconstruire des DDAs, il était nécessaire d'introduire une erreur de traitement. Le premier but était de visualiser son impact sur le résultat obtenu. Le second, était de montrer la capacité des algorithmes à détecter des erreurs produites par l'accélérateur de particules.

Pour cela, nous avons choisi de simuler une mauvaise position de lame. En effet, on a gardé une lame à sa position initiale alors qu'elle devait être en mouvement durant l'irradiation. Avant de décrire la simulation de la mauvaise position de lame, le type de traitement a été choisi. Il était plus pertinent de tester un traitement RCMI car les MLCs sont en mouvement durant l'irradiation. Afin d'atteindre cet objectif, nous avions besoin de déterminer l'angle de rotation du collimateur accessible depuis

l'image DICOM RT_Plan avec l'attribut « Beam Limiting Device Angle » catégorisé dans l'attribut « Control Point Sequence ». Possédant ces informations, un algorithme permettant de simuler une mauvaise position de lame a été implémenté sur Matlab®. Le résultat obtenu est visible sur la Figure 4.8. À Gauche, est présentée l'image EPID originale et à droite, l'image EPID simulée.

Afin d'avoir un signal simulé qui ressemble à la réalité, les valeurs attribuées à la lame positionnée anormalement ont été ajustées à des valeurs tout juste supérieures au threshold précédemment effectué. De plus, les valeurs de toute la lame ne sont pas identiques. Cette image simulée sera considérée uniquement dans la phase d'inférence. Une erreur de traitement doit être diagnostiquée pendant la phase de production (inférence), et aura une influence uniquement sur la mesure durant le traitement (imageur portal EPID). Cette simulation permettra de voir si le calcul de DDAs délivrée est cohérent et a pris en compte l'erreur provenant de l'accélérateur de particules. Le calcul de DDA a été effectué par l'intermédiaire des deux modèles précédemment exposés. Il sera intéressant de confronter les résultats obtenus par le réseau de neurones classique et le CNN.



(a) Image EPID originale (b) Image EPID simulée

FIGURE 4.8 – Comparaison entre image EPID originale et simulée pour la RCMI 6 MeV Varian® à partir d'un RNA classique

4.3 Résultats et discussions

Dans la partie précédente, ont été exposées les différentes méthodes utilisées pour le calcul de DDA délivrée durant la phase de prétraitement des patients. Deux modèles se sont avérés efficaces, un concerne un FFNN, l'autre concerne un CNN. Pour chacun de ces modèles, différents échantillons d'entrées ont été fournis en fonction de l'énergie de traitement utilisée. Différents types de traitements ont été abordés provenant de deux fabricants d'accélérateurs de particules. Dans cette section, seront abordés les résultats obtenus pour chacun des cas énoncés dans la section 4.2. La première partie sera consacrée aux résultats obtenus avec les réseaux de neurones classiques. La seconde partie est réservée aux résultats obtenus avec les CNNs. Ensuite, une évaluation des γ_{index} sera faite pour les différents traitements. Puis, pour finir une comparaison des résultats des deux modèles sera observée.

4.3.1 Résultats obtenus à partir des réseaux de neurones classiques

Les FFNNs ont été utilisés dans de nombreux cas. Ils ont été employés pour différents types de traitements, différentes énergies, et pour deux marques d'accélérateurs. Dans cette partie, les résultats obtenus via le modèle RNA classique pour chacune des applications sont présentés. L'ordre de présentation correspond à l'ordre chronologique dans lequel les résultats ont été produits. Les tout premiers ont été obtenus avec des traitements de radiothérapie conformationnelle. Ensuite, les algorithmes ont été étendus à la radiothérapie conformationnelle à modulation d'intensité. Une vérification de la prise en considération d'une erreur machine a été abordée avant d'étendre les algorithmes sur une énergie de traitement différente. Pour finir, la présentation des résultats obtenus pour un accélérateur de particules Elekta® de marque différente est effectuée. À la fois les traitements conformationnels et à modulation d'intensité ont été approfondis.

4.3.1.1 Pour la radiothérapie conformationnelle avec un accélérateur Varian®

La RC a été brièvement exposée en 2.3.1. On a pu voir que la particularité de ce type de traitement était sa simplicité. En effet, le bras de l'accélérateur et le collimateur multi-lames restent statiques durant l'irradiation. Cela implique un champ d'irradiation relativement homogène tel qu'on peut l'apercevoir sur la mesure EPID montrée en Figure 4.9, représentant un cerveau entier. Cette mesure a été faite dans les mêmes conditions qu'exposées en 2.2.5 et en 4.2.1.2.1. En effet, la phase de prétraitement a été faite sans objet sur la table et avec l'EPID à sa position de traitement (à 150 cm de la tête de l'accélérateur).

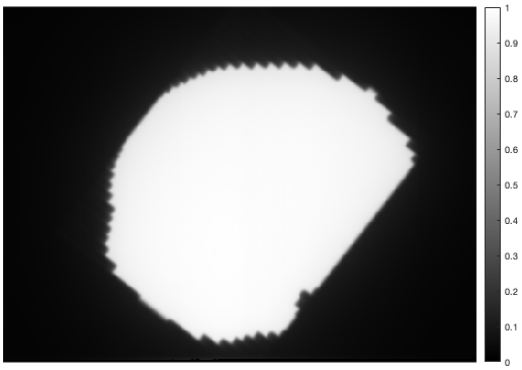


FIGURE 4.9 – Image EPID pour la RC 6 MeV Varian® à partir d'un RNA classique

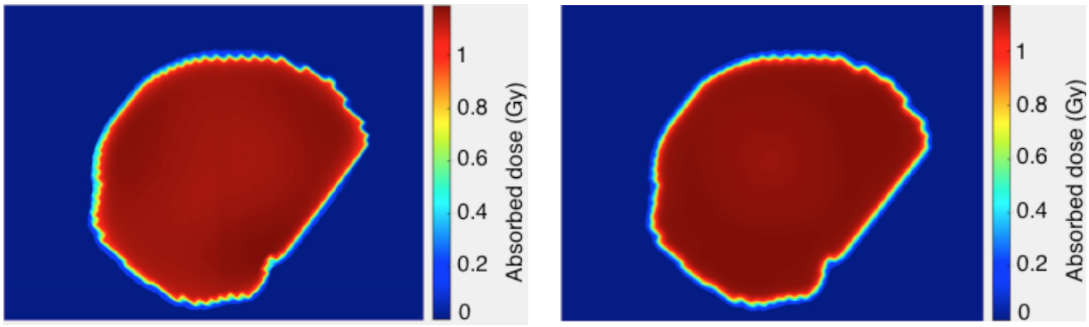
La phase d'apprentissage pour ce type de traitement a été effectuée avec sept ensembles de données d'entrées/sorties. Nous pouvons voir sur la Figure 4.10a, la DDA calculée par le réseau de neurones et sur la Figure 4.10b, la DDA originalement planifiée. Le traitement considéré durant la phase d'inférence, concerne la phase de prétraitement et l'irradiation d'un cerveau entier impliquant la présence de grands champs. On peut

s'apercevoir visuellement que les DDAs sont relativement proches. On remarque que la DDA calculée est en valeur absolue. Ces valeurs ont été obtenues par l'intermédiaire de la méthode présentée en 4.2.2.

Comme exposé en 4.2.4, le critère permettant d'évaluer la qualité des traitements est le γ_{index}. Un $\gamma_{index}(2\%,\ 2\ mm)$ global de 99,96% a été obtenu montrant la capacité des réseaux de neurones à calculer une DDA délivrée à partir d'un imageur portal EPID.

Ces premiers résultats obtenus nous ont montré que les algorithmes d'apprentissage automatique pouvaient être utilisés pour reconstruire des DDAs délivrées pour la RC.

La deuxième étape concerne l'extension des algorithmes afin de pouvoir calculer des DDAs délivrées pour des traitements de RCMI.



(a) DDA calculée (b) DDA planifiée

FIGURE 4.10 – Comparaison entre image calculée et planifiée pour la RC 6 MeV Varian® à partir d'un RNA classique

4.3.1.2 Pour la radiothérapie conformationnelle à modulation d'intensité avec un accélérateur Varian®

On a vu en 2.3.2, que la RCMI était une forme évoluée de la RC. Cette fois-ci, le collimateur multi-lames est dynamique durant l'irradiation permettant d'avoir une DDA hétérogène en terme de signal sur deux dimensions. Le bras de l'accélérateur de particules reste statique pendant le traitement. Le traitement considéré ici, est de type « sliding windows » et possède une séquence continue d'irradiation. En regardant la Figure 4.11, on constate le signal hétérogène obtenu. Tout comme le traitement précédent, l'EPID était placé à 150 cm de la tête de l'accélérateur de particules et aucun objet n'était placé sur la table pendant la phase de prétraitement.

La phase d'apprentissage pour ce type de traitement a été effectuée avec neuf ensembles de données d'entrées/sorties. Les Figures 4.12a et 4.12b montrent respectivement, la DDA calculée par le réseau de neurones et la DDA originalement planifiée. Le traitement considéré durant la phase d'inférence, concerne la phase de prétraitement d'un O.R.L impliquant la présence de champs de taille moyenne. On remarque une nouvelle fois que les DDAs obtenues sont relativement proches. La DDA calculée est de nouveau en valeur absolue. Cette fois, un $\gamma_{index}(2\%,\ 2\ mm)$ global de 97,7% a été obtenu. Ce résultat est quasiment aussi élevé que le précédent, montrant, de nouveau,

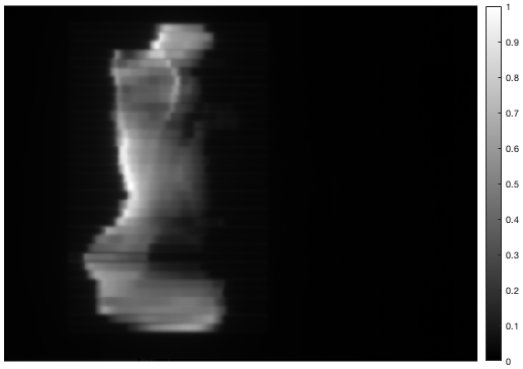
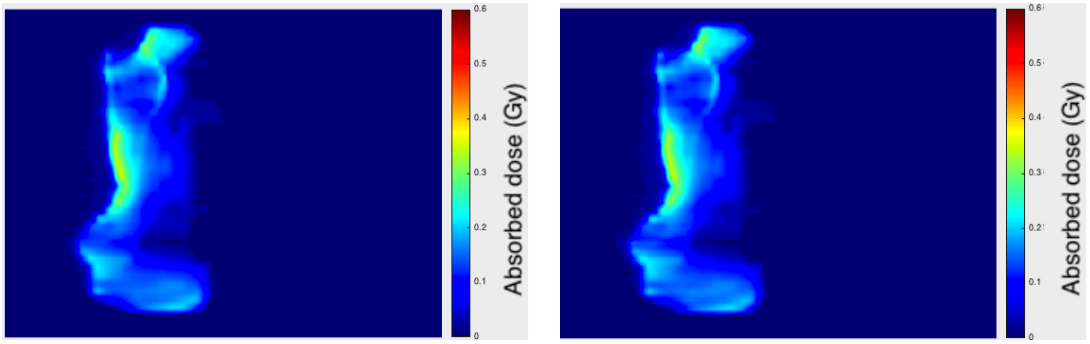


FIGURE 4.11 – Image EPID pour la RCMI 6 MeV Varian® à partir d'un RNA classique

la capacité des réseaux de neurones à calculer une DDA délivrée à partir d'un imageur portal EPID, cette fois-ci, pour un traitement RCMI.



(a) DDA calculée (b) DDA planifiée

FIGURE 4.12 – Comparaison entre image calculée et planifiée pour la RCMI 6 MeV Varian® à partir d'un RNA classique

Il a été apprécié de constater durant cette étape, la capacité des algorithmes à s'adapter en fonction des données transmises. En effet, les résultats ont été obtenus à partir du même modèle. Pour autant, les traitements considérés sont fondamentalement différents.

L'objectif de l'étape suivante sera de montrer la pertinence de notre modèle pour reconstruire une DDA malgré la présence d'une erreur machine simulée. Revenons auparavant au prétraitement RCMI appliqué sans erreur.

4.3.1.3 Pour la radiothérapie conformationnelle à modulation d'intensité avec une erreur de traitement simulée

Maintenant que les premiers résultats obtenus ont été montrés, quelques études montrant l'efficacité de l'apprentissage vont être abordées. Une fois que le réseau de neurones a été paramétré en fonction des données fournies, il était important de vérifier que le modèle obtenu était correctement mis à l'échelle et fixé. Pour cela, les valeurs de performance et de régression durant la phase d'apprentissage, ont été évaluées

pour chacun des cas traités dans ce manuscrit. La Figure 4.14a montre que pour l'apprentissage de ce type de traitement, 652 epochs ont été effectuées et que la fonction coût décroit avec une allure exponentiellement décroissante avant d'atteindre une valeur environnant 4×10^{-4}. On remarque aussi que proche des 300 epochs, la MSE a presque atteint sa valeur finale. De plus, on peut voir que les courbes de performances obtenues pour les données réservées à l'entrainement, le test et la validation se superposent montrant que le modèle a été correctement calibré. Cela signifie que les données ont été correctement mises à l'échelle et que le réseau de neurones a été capable de généraliser le modèle à de nouvelles données.

Sur la Figure 4.14b, on peut observer la ligne de régression entre la cible TPS et la valeur prédite par le réseau de neurones obtenu durant la phase d'apprentissage. Cette ligne représente à la fois les données réservées à l'entrainement, au test et à la validation. Cette métrique permet d'évaluer la qualité de la phase d'apprentissage ayant été effectuée. La valeur du coefficient de régression obtenue pour ce traitement était de 0,997 pour approximativement un million d'échantillons de données. La valeur du coefficient recherché est de 1 pour un apprentissage parfait. Dans cet exemple de traitement, cela signifie que beaucoup de valeurs de pixels prédites par le réseau de neurones sont proches de la valeur cible. Ces résultats obtenus montrent la cohérence trouvée entre les données d'entrées et de sorties des algorithmes. Ils montrent aussi que le modèle de réseau de neurones créé est hautement représentatif du comportement du système de traitement.

De plus, sur la Figure 4.13, apparaissent les lignes de régression pour chaque ensemble de données de la phase d'apprentissage. On peut y voir les résultats obtenus depuis les données d'entrainement, de test et de validation. Si on les regarde de plus près, on remarque que les valeurs de régression obtenues pour le test et la validation sont proches de celle d'entrainement. Cela signifie que le modèle réussit à généraliser ses résultats à de nouveaux échantillons de données montrant que le modèle représente correctement le système étudié. Pour la suite, seule la ligne de régression de l'ensemble des données est montrée. Cependant, la même dynamique pour les phases de test et validation a été obtenue.

La phase d'apprentissage a montré ci-dessus que le modèle créé était généralisable à de nouvelles données. Pour le vérifier, on a utilisé un échantillon de données d'entrées/-sorties qui n'avait pas été préalablement appris par les algorithmes. L'image EPID utilisée durant la phase d'inférence apparait sur la Figure 4.15. On remarque que le collimateur multi-lames n'a pas son angle de rotation par défaut. Les conditions de mesures de prétraitement sont les mêmes qu'étudiées auparavant.

La phase d'apprentissage pour ce type de traitement a été effectuée avec dix ensembles de données d'entrées/sorties. Le modèle reste identique à celui montré auparavant, seuls les poids ont été mis à jour en fonction des données transmises à l'algorithme. Les Figures 4.16a et 4.16b montrent respectivement, la DDA calculée par le réseau de neurones et la DDA planifiée par le TPS. Le traitement considéré durant la phase d'inférence est un traitement encéphalique. La DDA calculée est en valeur absolue. Cette fois, un $\gamma_{index}(2\%,\ 2\ mm)$ global de 98,2% a été obtenu. Un score restant

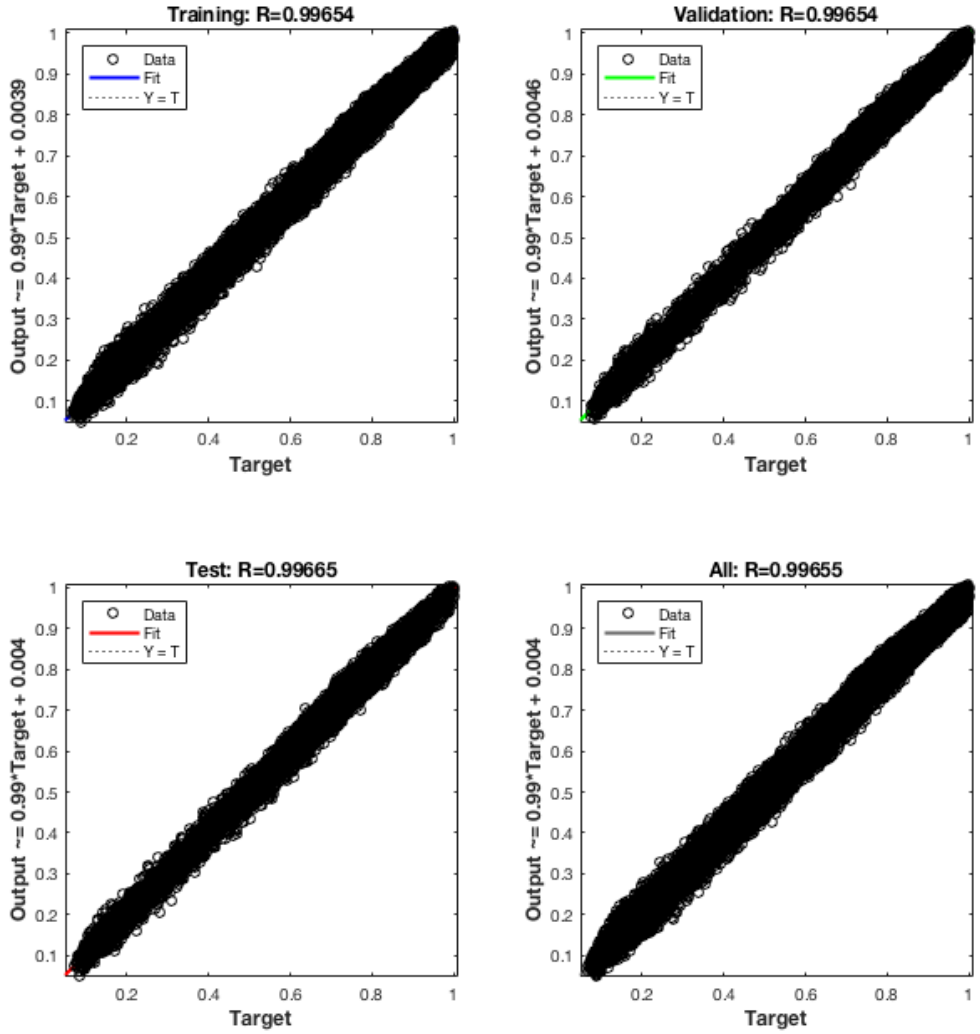
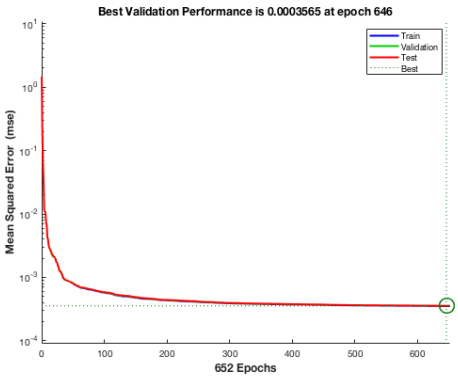


FIGURE 4.13 – Régression des différents ensembles de données de la phase d'apprentissage

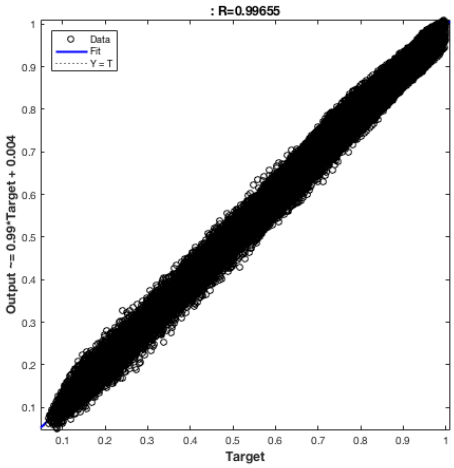
élevé pour ce type de traitement.

La carte des γ est visualisable sur la Figure 4.17. Cette carte permet de visualiser les endroits qui ne passent pas le critère $\gamma_{index}(2\%,\ 2\ mm)$. On voit que ce nombre de points est très faible et que l'accélérateur a délivré à peu de chose près, la dose absorbée planifiée. La prochaine étape consiste à vérifier la capacité des algorithmes à détecter une erreur durant le traitement.

Dans le cas d'un traitement identique à celui vu précédemment, l'erreur de traitement a été simulée par une mauvaise position de lame. Pour cela, la simulation d'une lame du MLC restée à sa position initiale durant le traitement a été effectuée. L'influence simulée sur l'EPID est montrée sur la Figure 4.18a. La DDA obtenue via les réseaux de



(a) Courbe de performance de la phase d'apprentissage



(b) Régression de la phase d'apprentissage

FIGURE 4.14 – Performance et régression de l'apprentissage pour la RCMI 6 MeV Varian® à partir d'un RNA classique

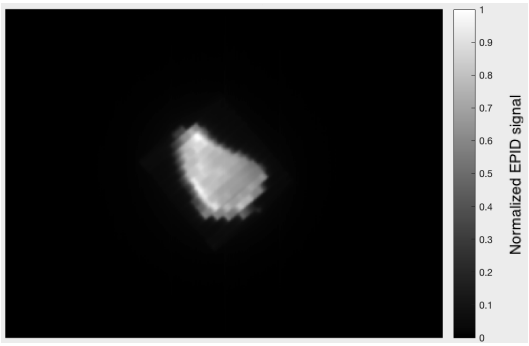
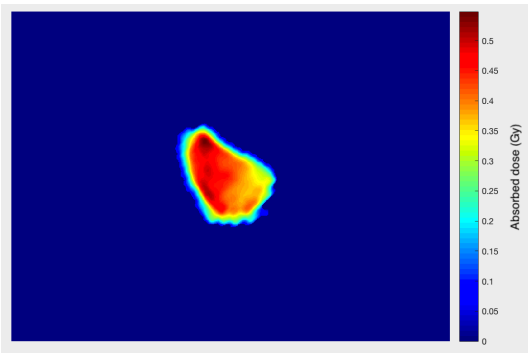
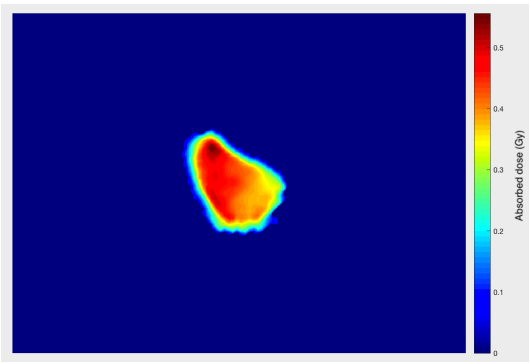


FIGURE 4.15 – Image EPID pour la RCMI 6 MeV Varian® à partir d'un RNA classique



(a) DDA calculée



(b) DDA planifiée

FIGURE 4.16 – Comparaison entre image calculée et planifiée pour la RCMI 6 MeV Varian® à partir d'un RNA classique

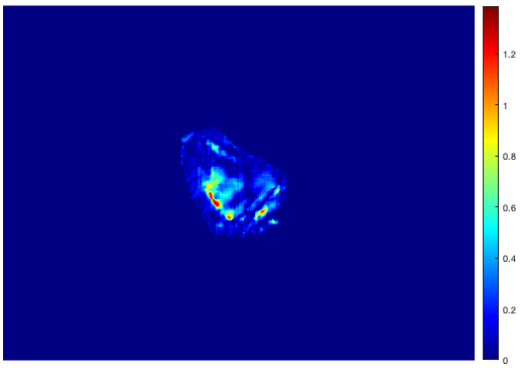
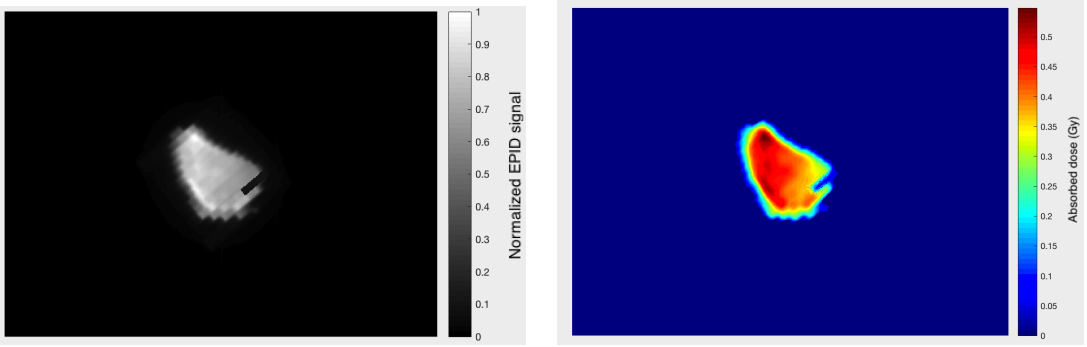


FIGURE 4.17 – Carte des $\gamma_{index}(2\%,\ 2\ mm)$ pour la RCMI 6 MeV Varian® à partir d'un RNA classique

neurones classiques est visible sur la Figure 4.18b. Elle doit être comparée à la DDA planifiée qui apparait en Figure 4.16b. On remarque rapidement l'influence qu'a pu avoir la mauvaise position de lame durant le traitement sur la DDA calculée. Ce qui apparait intéressant, malgré un apprentissage sans erreur de traitement (aucun cas de mauvaise position de lame dans les données d'apprentissage), l'algorithme fournit une DDA cohérente. De plus, une modélisation de la diffusion semble avoir été prise en compte puisqu'on remarque un contraste moins net au niveau des contours de la DDA prédite plutôt que sur l'image EPID. Un $\gamma_{index}(2\%,\ 2\ mm)$ global de 94% a été obtenu, quasiment 4,2% de différence avec celui précédemment calculé. Ces 4,2% correspondent à la représentation spatiale qu'occupe la lame comparée à l'ensemble de l'image. Cela montre la capacité des algorithmes à détecter une erreur produite durant le traitement.

Jusqu'à présent seul les résultats obtenus pour des traitements avec une énergie de 6 MeV ont été présentés. La prochaine étape concerne l'extension des algorithmes afin de pouvoir calculer des DDAs délivrées pour des traitements de RCMI avec une énergie de 25 MeV.



(a) Image EPID Simulée (b) DDA calculée

FIGURE 4.18 – Simulation d'une mauvaise position de lame pour la RCMI 6 MeV Varian® à partir d'un RNA classique

4.3.1.4 Pour la radiothérapie conformationnelle à modulation d'intensité avec un accélérateur Varian® et une énergie de 25 MeV

L'objectif de cette étape était de voir si les algorithmes pouvaient être étendus à de nouvelles énergies de traitement. Pour cela, une base de données de traitement avec une énergie de 25 MeV a été récupérée.

Le traitement considéré pour cette étape est la RCMI. La séquence de traitement est continue de type « sliding window » En regardant la Figure 4.19, on constate une nouvelle fois, le signal hétérogène obtenu. Tout comme les traitements précédents, l'EPID était placé à 150 cm de la tête de l'accélérateur de particules et aucun objet n'était placé sur la table pendant la phase de prétraitement.

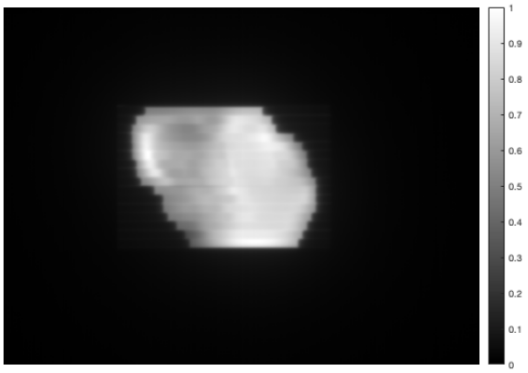
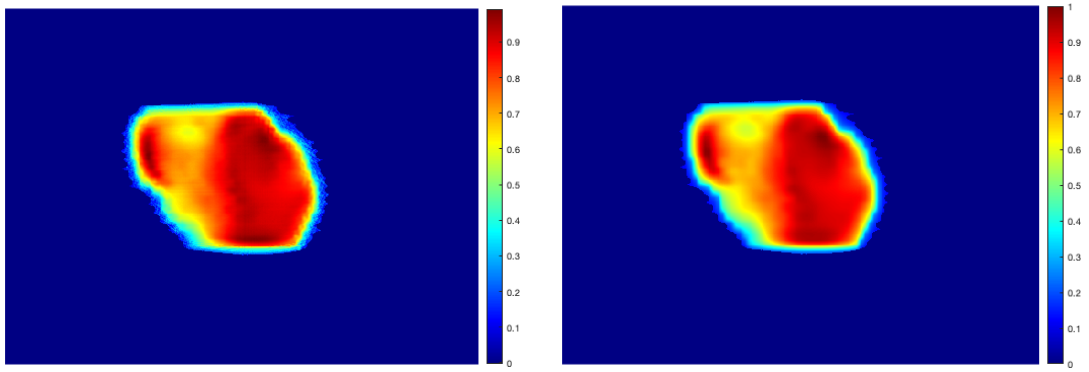


FIGURE 4.19 – Image EPID pour la RCMI 25 MeV Varian®à partir d'un RNA classique

La phase d'apprentissage pour ce traitement a été effectuée avec quatre ensembles de données d'entrées/sorties. Les Figures 4.20a et 4.20b montrent respectivement, la DDA prédite par le réseau de neurones et la DDA originalement planifiée. Le traitement considéré durant la phase d'inférence, concerne la phase de prétraitement d'une prostate impliquant la présence de champs de taille moyenne. On remarque une nouvelle fois que les DDAs obtenues sont relativement proches. Cependant, cette fois-ci, il a été indiqué en 4.2.3 que le nombre d'entrées différait d'habituellement. En effet, étant dans la situation où l'énergie est plus élevée, la diffusion se propage sur un rayon plus grand. La nécessité de considérer un plus grand nombre d'entrées est apparue pour obtenir des DDAs calculées optimales.

La DDA calculée est cette fois-ci, à valeur relative. Il faudrait lui appliquer le coefficient de dénormalisation vu en 4.2.2 pour obtenir une DDA en valeur absolue. Un $\gamma_{index}(2\%,\ 2\ mm)$ global de 99,91% a été obtenu. Ce résultat reste très élevé, montrant une nouvelle fois, la capacité des réseaux de neurones à calculer une DDA délivrée à partir d'un imageur portal EPID, à différentes énergies.

Une nouvelle fois, nous avons pu constater durant cette étape, la capacité des algorithmes à s'adapter en fonction des données transmises. En ayant étudié les phénomènes physiques sous-jacents, le nombre d'entrées a pu être adapté comme montré en 4.2.3.1. Cette nouvelle énergie a été intégrée aux algorithmes. Dans le réseau de neurones, seul le nombre de neurones a varié permettant de produire une seule application traitant



(a) DDA calculée (b) DDA planifiée

FIGURE 4.20 – Comparaison entre image calculée et planifiée pour la RCMI 25 MeV Varian® à partir d'un RNA classique

tous les cas rencontrés jusqu'à présent.

L'objectif de l'étape suivante sera de montrer la possibilité d'étendre les algorithmes pour des nouveaux accélérateurs de particules de marques différentes.

4.3.1.5 RC Elekta®

Cette partie concerne un nouveau challenge. Il consistait à démontrer les performances des algorithmes sur des données issues d'un accélérateur de marque différente.

On remarque le champ d'irradiation relativement homogène sur la mesure EPID représentée en Figure 4.22. Cette mesure a été faite dans les mêmes conditions qu'exposées en 2.2.5 et en 4.2.1.2.2. En effet, la phase de prétraitement a été faite sans objet sur la table et avec l'EPID à sa position de traitement (à 160 cm de la tête de l'accélérateur).

La phase d'apprentissage pour ce type de traitement a été effectuée avec six ensembles de données d'entrées/sorties. Par comparaison avec un traitement de même type depuis un accélérateur Varian®, des modèles identiques ont été appliqués. Seuls les poids du modèle ont été mis à jour durant la phase d'apprentissage. Nous pouvons voir sur la Figure 4.23a, la DDA calculée par le réseau de neurones et sur la Figure 4.23b, la DDA originalement planifiée. Le traitement considéré durant la phase d'inférence, concerne la phase de prétraitement et l'irradiation d'un cerveau entier impliquant la présence de grands champs. On peut s'apercevoir visuellement que les DDA restent proches. On remarque que la DDA calculée est à valeur relative. Un $\gamma_{index}(2\%,\ 2\ mm)$ global de 98,5% a été obtenu montrant la capacité des réseaux de neurones à calculer une DDA délivrée à partir d'un imageur portal EPID Elekta®.

De plus, on remarque sur la Figure 4.21, que le coefficient de régression obtenu durant la phase d'apprentissage est de 0,997 pour approximativement un million d'échantillons de données. Dans cet exemple de traitement, cela signifie que beaucoup de valeurs de pixels prédits par le réseau de neurones sont proches de leur valeur cible. On remarque

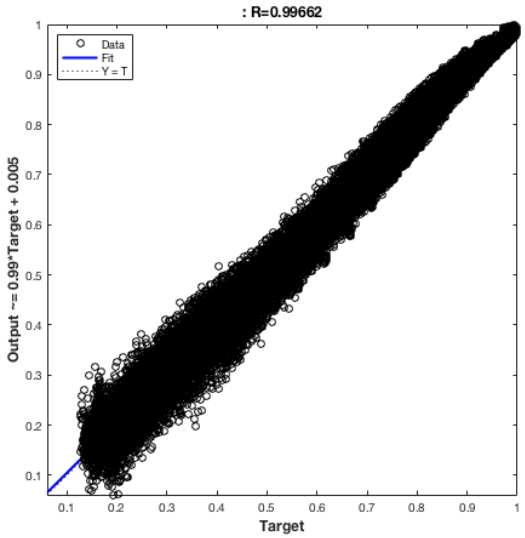


FIGURE 4.21 – Régression de la phase d'apprentissagepour la RC 6 MeV Elekta® à partir d'un RNA classique

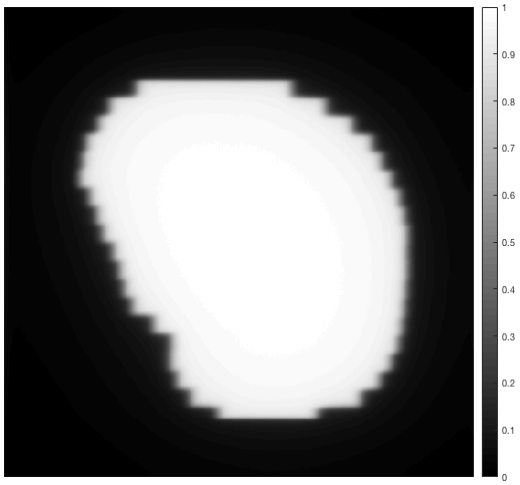


FIGURE 4.22 – Image EPID pour la RC 6 MeV Elekta® à partir d'un RNA classique

aussi que les écarts les plus importants se trouvent sur les valeurs à faible dose absorbée. Ces résultats obtenus montrent une nouvelle fois la cohérence trouvée entre les données d'entrées et de sorties des algorithmes.

Ces résultats obtenus nous ont montré que les algorithmes d'apprentissage automatique pouvaient être utilisés pour reconstruire des DDAs délivrées pour la RC à partir de différents accélérateurs de particules.

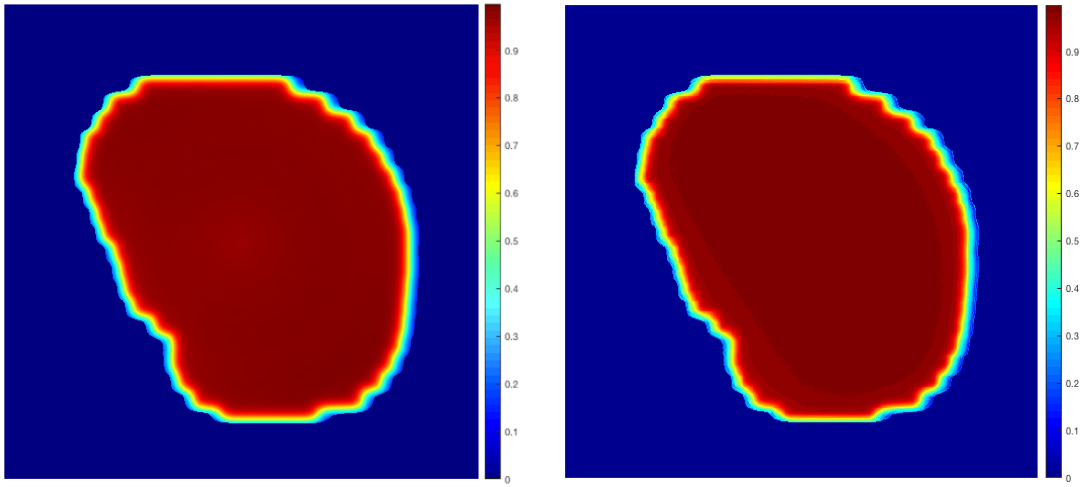
La prochaine étape concerne l'extension des algorithmes pour calculer des DDAs délivrées pour des traitements de RCMI à partir d'une machine Elekta®.

4.3.1.6 Elekta® RCMI

Le traitement considéré ici, est de type « step and shot » et possède une séquence discrète d'irradiation. En regardant la Figure 4.25, on constate le signal hybride en terme d'homogénéité obtenue. Tout comme le traitement précédent, l'EPID était placé à 160 cm de la tête de l'accélérateur de particules et aucun objet n'était placé sur la table pendant la phase de prétraitement.

La phase d'apprentissage pour ce type de traitement a été effectuée avec quatre ensembles de données d'entrées/sorties. Tout comme pour le type de traitement précédent, le modèle utilisé pour cette marque d'accélérateur est identique à celui utilisé pour l'accélérateur Varian®. Une nouvelle fois, les poids du modèle ont été actualisés en fonction des données transmises durant la phase d'apprentissage. Les Figures 4.26a et 4.26b montrent respectivement, la DDA calculée par le réseau de neurones et la DDA originalement planifiée.

Le traitement considéré durant la phase d'inférence, concerne la phase de prétraite-



(a) DDA calculée (b) DDA planifiée

FIGURE 4.23 – Comparaison entre image calculée et planifiée pour la RC 6 MeV Elekta® à partir d'un RNA classique

ment d'un cerveau entier, impliquant la présence de grands champs d'irradiation. On remarque une nouvelle fois que les DDAs obtenues sont relativement proches. La DDA calculée est de nouveau, à valeur relative. Cette fois, un $\gamma_{index}(2\%,\ 2\ mm)$ global de 98,11% a été obtenu.

Ce résultat est quasiment aussi élevé que le précédent, montrant, de nouveau, la capacité des réseaux de neurones à calculer une DDA délivrée à partir d'un imageur portal EPID Elekta®, cette fois-ci, pour un traitement RCMI.

De plus, on remarque sur la Figure 4.25, que le coefficient de régression obtenu durant la phase d'apprentissage est de 0,998 pour approximativement cinq cents milles échantillons de données. On remarque aussi une répartition homogène des écarts obtenus sur les faibles et fortes doses absorbées. Ces résultats obtenus montrent une nouvelle fois la cohérence trouvée entre les données d'entrées et de sorties des algorithmes.

Cette étape a permis de mettre en lumière, la capacité des algorithmes à s'adapter en fonction des données transmises. En effet, on a vu les résultats obtenus avec de très bons critères γ_{index} pour différents types de traitement, différentes énergies et différents accélérateurs de particules.

L'objectif de l'étape suivante sera de montrer la pertinence d'un nouveau modèle utilisé basé sur un CNN pour reconstruire une DDA.

4.3.2 Résultats obtenus à partir des CNNs

Les CNNs ont été utilisés pour différentes énergies. L'objectif de cette partie est de montrer la pertinence de ce type de réseau pour cette application de radiothérapie externe. Les modèles ont été établis à partir des bases de données utilisées précédem-

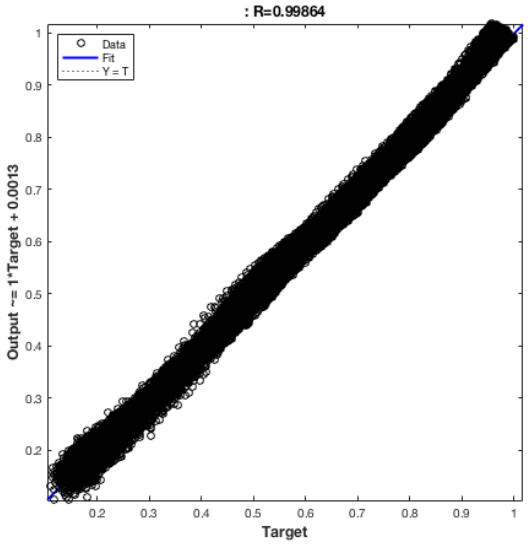


FIGURE 4.24 – Régression de la phase d'apprentissage pour la RCMI 6 MeV Elekta® à partir d'un RNA classique

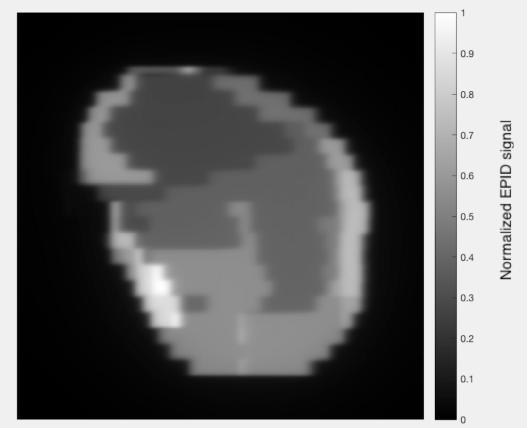
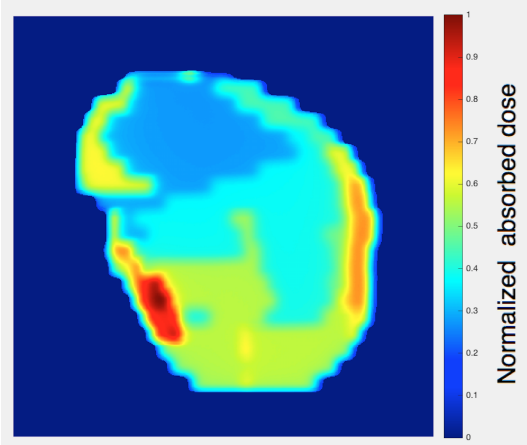
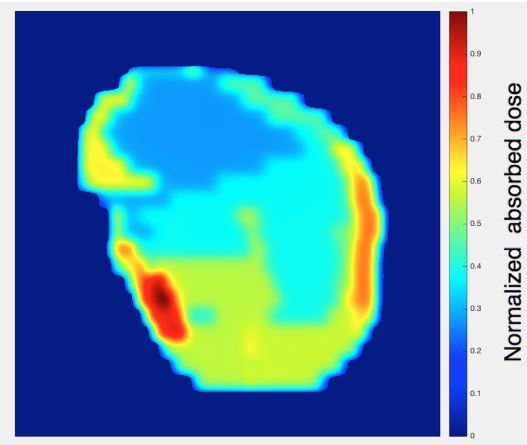


FIGURE 4.25 – Image EPID pour la RCMI 6 MeV Elekta® à partir d'un RNA classique



(a) DDA calculée



(b) DDA planifiée

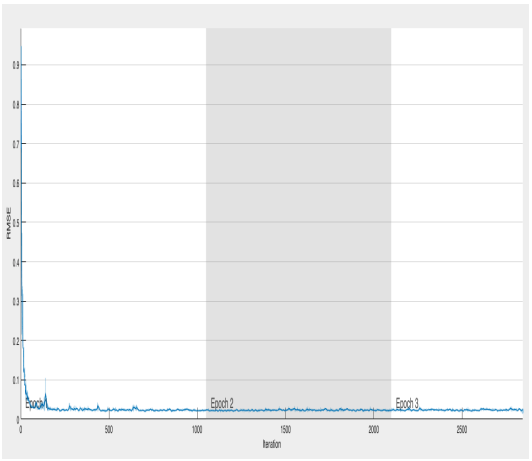
FIGURE 4.26 – Comparaison entre image calculée et planifiée pour la RCMI 6 MeV Elekta® à partir d'un RNA classique

ment avec les réseaux de neurones classiques. Des erreurs machines aux deux énergies considérées ont été simulées avec les mêmes conditions que pour les réseaux de neurones classiques.

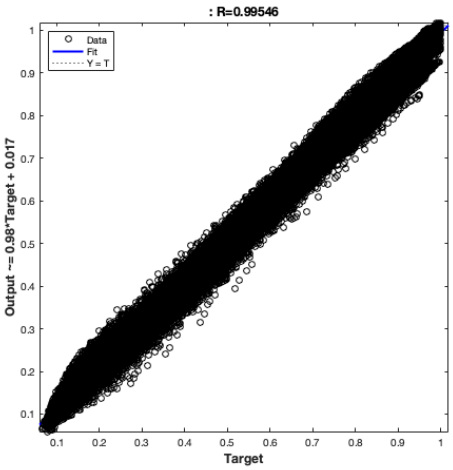
4.3.2.1 6 MeV

Maintenant que les résultats obtenus avec les réseaux de neurones classiques ont été montrés, on va aborder les résultats provenant des CNNs tels que décrits en 4.2.3.2. On remarque par l'intermédiaire de la Figure 4.27 que le modèle obtenu était correctement mis à l'échelle et paramétré. En effet, la Figure 4.27a montre que pour l'apprentissage

de ce type de traitement, 3 epochs ont été effectués et que la fonction coût décroit avec une allure exponentielle avant d'atteindre une valeur environnant $0,2$. On remarque aussi que proche des 300 itérations, la RMSE a presque atteint sa valeur finale. De plus, on note sur la Figure 4.27b, que le coefficient de régression atteint une valeur de 0,995, le plaçant proche du résultat obtenu à partir des réseaux de neurones classique. La régression obtenue représente simultanément les valeurs obtenues durant l'apprentissage avec les données réservées à l'entrainement, au test et à la validation.



(a) Courbe de performance de la phase d'apprentissage



(b) Régression de la phase d'apprentissage

FIGURE 4.27 – Performance et régression de l'apprentissage pour la RCMI 6 MeV Varian® à partir d'un CNN

Cette métrique permet d'évaluer la qualité de la phase d'apprentissage ayant été effectuée. On note une nouvelle fois qu'un grand nombre de valeurs de pixels prédites par le CNN sont proches de leur valeur cible. Ces résultats obtenus montrent que le modèle de CNN créé est hautement représentatif du comportement du système de traitement.

La phase d'apprentissage a montré ci-dessus que le modèle créé était général à de nouvelles données. Pour le vérifier, on a utilisé un échantillon de données d'entrées/sorties identique à celui utilisé avec les réseaux de neurones classiques. L'image EPID utilisée durant la phase d'inférence apparait sur la Figure 4.28.

La phase d'apprentissage pour ce type de traitement a été effectué avec dix ensembles de données d'entrées/sorties. Les Figures 4.29a et 4.29b montrent respectivement, la DDA calculée par le CNN et la DDA absorbée planifiée par le TPS. Le traitement considéré durant la phase d'inférence est un traitement encéphalique. La DDA calculée est en valeur absolue. Cette fois, un $\gamma_{index}(2\%,\ 2\ mm)$ global de 98,71% a été obtenu. On remarque un score légèrement inférieur à celui obtenu avec les réseaux de neurones classiques.

La carte des γ est représentée sur la Figure 4.30. Cette carte permet de visualiser

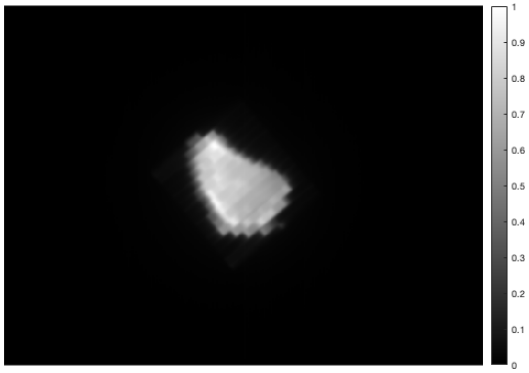
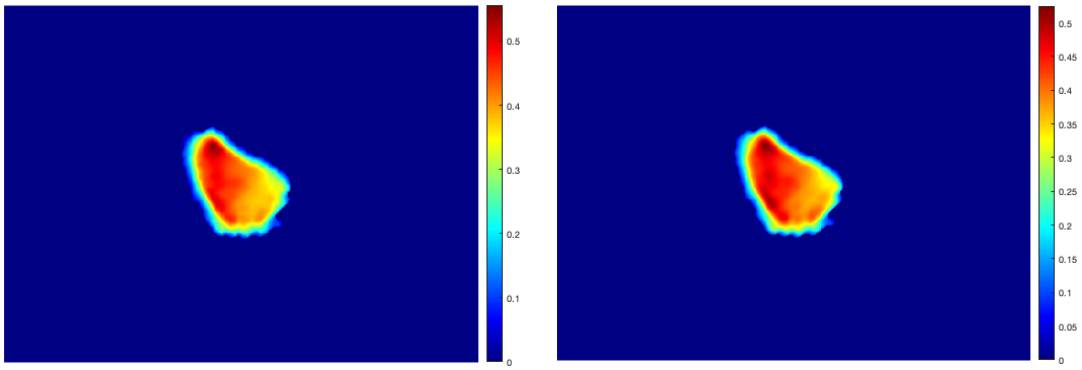


FIGURE 4.28 – Image EPID pour la RCMI 6 MeV Varian® à partir d'un CNN



(a) DDA calculée (b) DDA planifiée

FIGURE 4.29 – Comparaison entre image calculée et planifiée pour la RCMI 6 MeV Varian® à partir d'un CNN

les endroits qui ne passent pas le critère $\gamma_{index}(2\%,\ 2\ mm)$. On voit que ce nombre de points est très faible et que l'accélérateur a quasiment délivré, la dose absorbée planifiée. Un point important qu'il faut souligner est que les zones qui n'atteignent pas le critère, sont identiques à celles observées avec les réseaux de neurones classiques sur la Figure 4.17. La prochaine étape reprend la mauvaise position de lame simulée sur l'EPID visualisable sur la Figure 4.31a.

La DDA obtenue via les CNNs est visible sur la Figure 4.31b. On remarque l'influence qu'a pu avoir la mauvaise position de lame durant le traitement sur la DDA calculée. On remarque donc que, malgré un apprentissage sans erreur de traitement (aucun cas de mauvaise position de lame dans les données d'apprentissage), l'algorithme fournit une DDA cohérente. De plus, une modélisation de la diffusion semble avoir été prise en compte puisqu'on remarque un contraste moins net au niveau des contours de la DDA prédite plutôt que sur l'image EPID. Il est à noter une modélisation du diffusé bien plus précise avec ce modèle de CNN comparé à celui des réseaux de neurones classiques visible sur la Figure 4.18b. Un $\gamma_{index}(2\%,\ 2\ mm)$ global de 93,54% a été obtenu, quasiment 5% de différence avec celui précédemment calculé. Ces 5% correspondent à la représentation spatiale qu'occupe la lame comparé à l'ensemble de l'image. Cela montre la capacité des algorithmes à détecter une erreur produite durant le traitement.

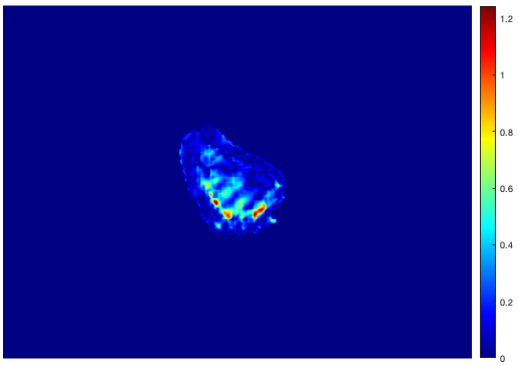
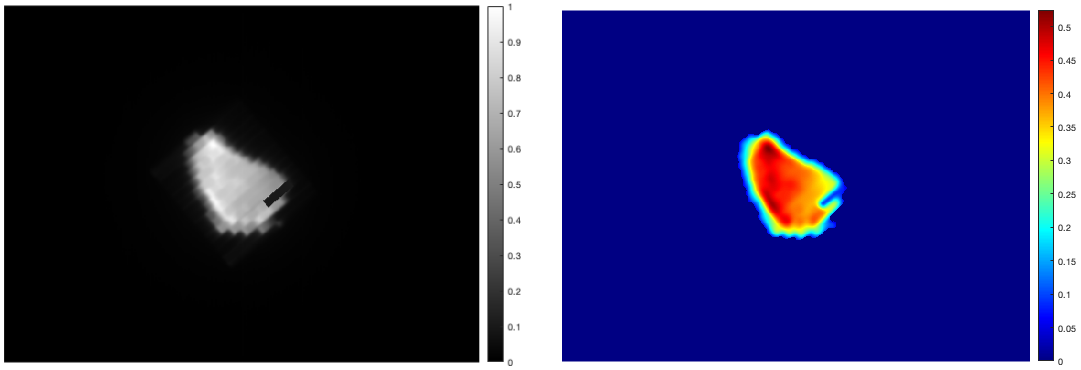


FIGURE 4.30 – Carte des $\gamma_{index}(2\%,\ 2\ mm)$ pour la RCMI 6 MeV Varian® à partir d'un CNN



(a) Image EPID Simulée (b) DDA calculée

FIGURE 4.31 – Simulation d'une mauvaise position de lame pour la RCMI 6 MeV Varian® à partir d'un CNN

La prochaine étape concerne l'extension des algorithmes afin de pouvoir calculer des DDAs délivrées pour des traitements de RCMI avec une énergie de 25 MeV à partir d'un CNN.

4.3.2.2 25 MeV

Les résultats obtenus avec les CNNs pour des traitements avec une énergie de 6 MeV ont été montrés dans la partie précédente. On a pu voir que ce type de modèle était adapté pour une reconstruction de DDA. Dans cette partie, nous allons aborder les résultats obtenus pour une énergie de traitement à 25 MeV.

Pour vérifier que les algorithmes de CNN étaient efficaces à différentes énergies, on a utilisé un échantillon de données d'entrées/sorties identique à celui utilisé avec les réseaux de neurones classiques. L'image EPID utilisée durant la phase d'inférence apparait sur la Figure 4.32.

La phase d'apprentissage pour ce type de traitement a été effectuée avec quatre ensembles de données d'entrées/sorties. L'énergie considérée étant différente du cas

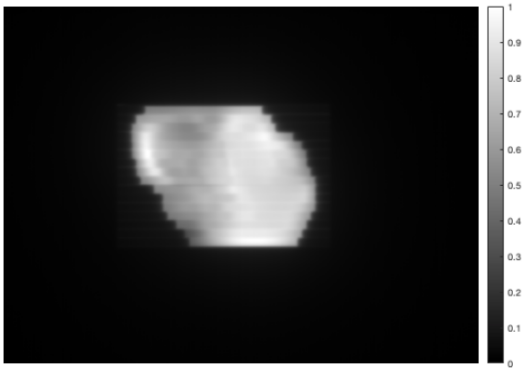
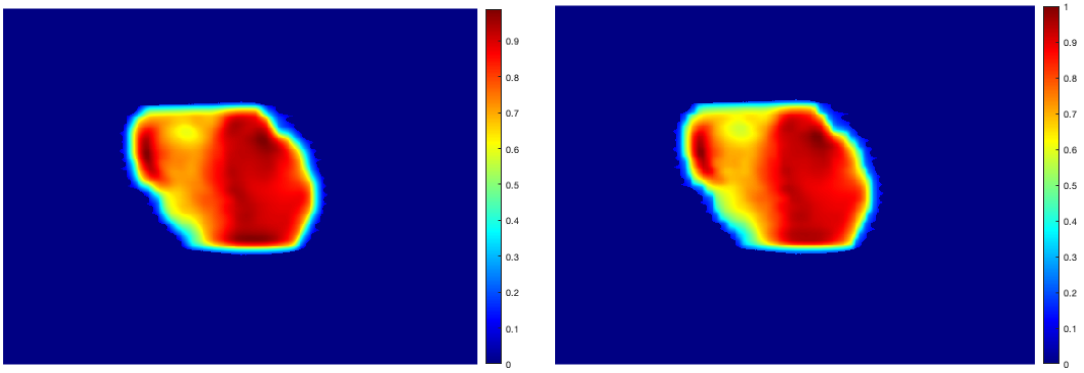


FIGURE 4.32 – Image EPID pour la RCMI 25 MeV Varian® à partir d'un CNN

traité précédemment, la taille des filtres à convolution diffère comme montré en 4.2.3.2. Cependant, le nombre de filtres reste inchangé et le reste du modèle est équivalent. Les Figures 4.33a et 4.33b montrent respectivement, la DDA calculée par le CNN et la DDA planifiée par le TPS. Le traitement considéré durant la phase d'inférence est un traitement de prostate. La DDA calculée est à valeur relative. Cette fois, un $\gamma_{index}(2\%,\ 2\ mm)$ global de 99,66% a été obtenu. On remarque un score légèrement inférieur à celui obtenu avec les réseaux de neurones classiques.



(a) DDA calculée (b) DDA planifiée

FIGURE 4.33 – Comparaison entre image calculée et planifiée pour la RCMI 25 MeV Varian® à partir d'un CNN

La carte des γ est donnée sur la Figure 4.34. Elle permet de voir les endroits qui ne respectent pas le critère $\gamma_{index}(2\%,\ 2\ mm)$.

On remarque que ce nombre de points est de nouveau très faible avec des zones qui n'atteignent pas le critère à faible dose absorbée. La prochaine étape reprend la mauvaise position de lame simulée sur l'EPID visualisable sur la Figure 4.35a.

La DDA obtenue via les CNNs est visible sur la Figure 4.35b. Ce qui apparait intéressant une nouvelle fois, malgré un apprentissage sans erreur de traitement (aucun cas de mauvaise position de lame dans les données d'apprentissage), est que l'algorithme fournit une DDA cohérente. Le même constat que pour une énergie de 6 MeV peut être

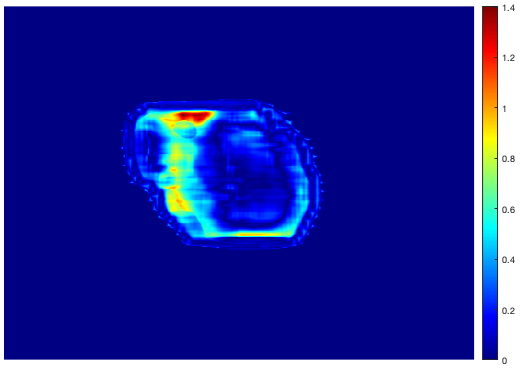
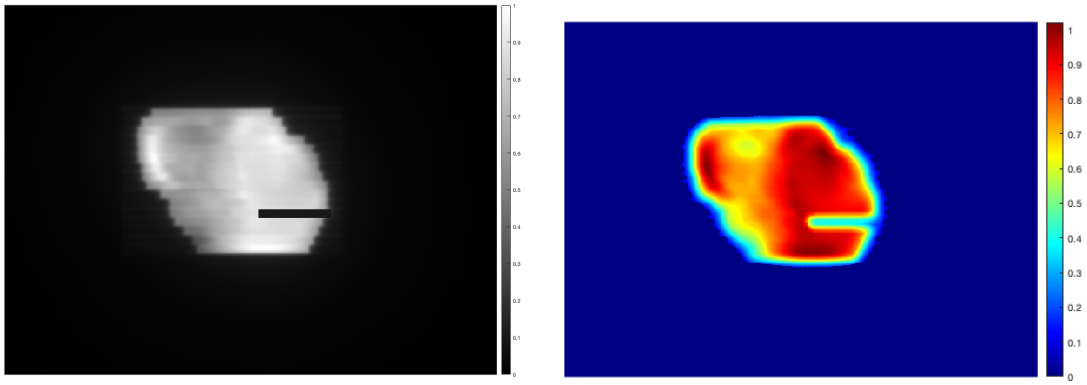


FIGURE 4.34 – Carte des $\gamma_{index}(2\%,\ 2\ mm)$ pour la RCMI 25 MeV Varian® à partir d'un CNN



(a) Image EPID Simulée (b) DDA calculée

FIGURE 4.35 – Simulation d'une mauvaise position de lame pour la RCMI 25 MeV Varian® à partir d'un CNN

fait, puisqu'on remarque une nouvelle fois un contraste moins net au niveau des contours de la DDA prédite comparée à celle sur l'image EPID. Cela montre, de nouveau, la capacité des algorithmes à détecter une erreur produite durant le traitement de 25 MeV à partir d'un CNN. La prochaine partie abordera la statistique obtenue pour chacun des modèles et pour les différentes configurations considérées.

4.3.3 Analyse et validations des modèles

Cette section est consacrée à l'analyse des résultats obtenus sur les différentes configurations étudiées auparavant. Le Tableau 4.1, récapitule les résultats acquis durant la phase de contrôle qualité. Les résultats affichés correspondent aux différents scores de γ_{index}. Pour chacune des configurations, une validation croisée a été faite à partir de l'ensemble des données présentes. Autrement dit, pour chaque cas, on a retiré une donnée pour la phase d'inférence et le reste appartenait à la base de données d'apprentissage. Chacune des données a été considérée pour la phase d'inférence avant d'être utilisée dans la base d'apprentissage du prochain set. Chaque apprentissage a été

		Nombre Images	γ_{index}(1%, 1 mm)				γ_{index}(2%, 2 mm)				γ_{index}(3%, 3 mm)			
			Global		Local		Global		Local		Global		Local	
			$\mathcal{P}_{\gamma<1}$(%)	γ_{moyen}	$\mathcal{P}_{\gamma<1}$(%)	γ_{moyen}	$\mathcal{P}_{\gamma<1}$(%)	γ_{moyen}	$\mathcal{P}_{\gamma<1}$(%)	γ_{moyen}	$\mathcal{P}_{\gamma<1}$(%)	γ_{moyen}	$\mathcal{P}_{\gamma<1}$(%)	γ_{moyen}
réseaux classiques	RC 6 MeV Varian®	7	89,27 ± 2,09	0,533 ± 0,022	87,28 ± 2,72	0,538 ± 0,031	99,97 ± 0,02	0,267 ± 0,012	99,96 ± 0,02	0,27 ± 0,011	99,998 ± 0,002	0,1787 ± 0,008	99,98 ± 0,007	0,181 ± 0,007
	RCMI 6 MeV Varian®	9	87,6 ± 1,12	0,4874 ± 0,15	83,83 ± 1,32	0,5404 ± 0,011	97,63 ± 0,82	0,2531 ± 0,008	96,67 ± 0,61	0,287 ± 0,01	99,86 ± 0,14	0,1703 ± 0,006	99,55 ± 0,21	0,195 ± 0,052
	RCMI 25 MeV Varian®	4	90,97 ± 2,46	0,4194 ± 0,04	88,64 ± 1,44	0,4624 ± 0,0261	99,94 ± 0,04	0,2112 ± 0,016	99,39 ± 0,6	0,235 ± 0,016	100 ± 0	0,142 ± 0,011	100 ± 0	0,159 ± 0,012
	RC 6 MeV Elekta®	6	88,6 ± 1,12	0,474 ± 0,039	84,1 ± 2,88	0,516 ± 0,051	98,43 ± 0,286	0,239 ± 0,012	98,03 ± 0,321	0,267 ± 0,014	99,93 ± 0,04	0,162 ± 0,007	99,69 ± 0,106	0,189 ± 0,008
	RCMI 6 MeV Elekta®	4	86,92 ± 1,4	0,53 ± 0,049	83,4 ± 2,81	0,525 ± 0,055	97,86 ± 0,358	0,28 ± 0,017	96,93 ± 0,43	0,287 ± 0,016	99,81 ± 0,132	0,181 ± 0,01	99,6 ± 0,111	0,194 ± 0,011
réseaux convolutionnels	RC 6 MeV Varian®	7	89,6 ± 2,57	0,504 ± 0,031	89,21 ± 2,41	0,51 ± 0,033	99,99 ± 0,01	0,253 ± 0,016	99,91 ± 0,09	0,256 ± 0,015	100 ± 0	0,169 ± 0,017	100 ± 0	0,172 ± 0,01
	RCMI 6 MeV Varian®	9	86,4 ± 2,37	0,512 ± 0,052	82,9 ± 3,1	0,567 ± 0,061	98,84 ± 0,36	0,259 ± 0,026	97,4 ± 0,61	0,293 ± 0,037	99,99 ± 0,01	0,174 ± 0,019	99,69 ± 0,31	0,199 ± 0,021
	RCMI 25 MeV Varian®	4	84,28 ± 8,01	0,527 ± 0,134	81,01 ± 10,47	0,584 ± 0,206	99,75 ± 0,21	0,265 ± 0,096	98,89 ± 0,43	0,299 ± 0,103	100 ± 0	0,178 ± 0,064	99,98 ± 0,02	0,203 ± 0,071
	RC 6 MeV Elekta®	6	89,7 ± 1,09	0,468 ± 0,04	85,1 ± 2,26	0,515 ± 0,049	99,62 ± 0,121	0,186 ± 0,011	98,99 ± 0,29	0,244 ± 0,013	99,998 ± 0,002	0,121 ± 0,006	99,88 ± 0,007	0,151 ± 0,007
	RCMI 6 MeV Elekta®	4	88,4 ± 1,2	0,48 ± 0,043	83,9 ± 2,7	0,535 ± 0,055	98,19 ± 0,31	0,247 ± 0,014	97,5 ± 0,37	0,28 ± 0,017	99,91 ± 0,04	0,166 ± 0,009	99,61 ± 0,11	0,191 ± 0,01

Tableau 4.1 – Résultats obtenus pour les différents traitements

effectué 8 fois et seuls les 5 meilleurs résultats ont été gardés. La valeur affichée dans le tableau correspond à la valeur moyenne obtenue sur l'ensemble des données formant la base associée au max des deux extrémités. Les résultats obtenus reflètent la qualité de l'apprentissage effectué pour tous les types de traitement et de modèles étudiés.

4.3.4 Comparaisons des deux types de modèles

Dans ces travaux, deux modèles de réseaux de neurones différents ont été utilisés. L'un est ancestral, le modèle feed-forward présenté en 1.4.3 et en 4.2.3.1. Le second est bien plus récent, le CNN présenté en 1.4.4 et en 4.2.3.2. On peut voir dans le Tableau 4.1, que les résultats obtenus à partir de ces deux modèles sont proches. Cependant, en zoomant sur les Figures 4.18b et 4.31b au niveau de la lame défaillante, on aperçoit une modélisation du diffusé plus précise avec le CNN. Autrement dit, la prédiction obtenue au niveau de la lame défaillante est moins nette avec le réseau de neurones classiques. À juste titre, on remarque que le calcul de chaque pixel est indépendant avec ce type de modèle provoquant une variation plus grande entre deux pixels voisins.

En 4.2.4, était montré l'influence que pouvait avoir de fortes variations sur le critère γ_{index}. C'est la principale raison de l'obtention de meilleurs scores γ_{index} dans certains cas, avec les réseaux de neurones classiques comparés aux CNNs. Pour finir, l'architecture des CNNs permet structurelllement de faire des calculs en fonction des valeurs contenues dans les pixels voisins grâce aux filtres à convolution. Elle permet une meilleure modélisation du diffusé qui intrinsèquement est dépendant des contributions dosimétriques provenant des pixels voisins.

4.4 Conclusion

Ce chapitre a abordé les premiers modèles d'apprentissage appliqués à la phase de contrôle qualité des traitements. Un éventail de traitements a été considéré, permettant de montrer la qualité de généralisation des algorithmes vers de nouveaux accélérateurs de particules notamment. Il a été montré dans cette partie que les algorithmes étaient capables de prédire une DDA pour deux accélérateurs de particules différents, deux types de traitement que sont la RC et la RCMI, tous deux à différentes énergies, 6 MeV ou 25 MeV. Il a aussi abordé la fiabilité des algorithmes pour reconstruire une DDA en cas de défaillance machine. Pour cela, la simulation d'une mauvaise position de lame a été choisie. Les algorithmes ont réussi à prédire une DDA cohérente alors qu'aucune mauvaise position de lame n'avait été introduite dans la base de données d'apprentissage. Le chapitre suivant porte sur l'extension des algorithmes vers la prédiction de DDA *in-vivo*.

Chapitre 5

Dosimétrie *in-vivo* à partir des distances radiologiques

5.1 Introduction

La dosimétrie *in-vivo* permet d'évaluer la dose absorbée reçue par les personnes exposées aux rayonnements ionisants durant un traitement. C'est une métrique importante du processus de vérification car c'est cette évaluation qui permettra d'informer la qualité du traitement reçu par le patient. Réussir à reconstruire une DIV à partir de techniques de machine learning est un challenge conséquent tellement le nombre de paramètres à prendre en compte est grand.

Il a été montré dans le chapitre précédent, l'efficacité des algorithmes de machine-learning à reconstruire une DDA durant la phase de contrôle qualité du traitement. La difficulté supplémentaire dans la reconstruction de la DIV concerne la prise en considération de la géométrie du patient. La suite de densités hétérogènes traversées par les rayons a un impact direct sur la dose absorbée déposée. De plus, une influence forte de dépôt d'énergie intervient spatialement (phénomène de diffusion). Chaque patient ayant des suites de densités tissulaires diférentes, la prise en considération de chacune des composantes est complexe de manière analytique. Dans ce cadre, l'utilisation des algorithmes de machine-learning porte tout son intérêt.

Dans ce chapitre, nous verrons l'importance que peuvent avoir les distances radiologiques dans les algorithmes et le modèle mathématique utilisé pour calculer ces dernières. De plus, il sera montré avec quelle efficacité CUDA [202] permettra d'obtenir un grand nombre de distances radiologiques en peu de temps, puis les méthodes algorithmiques associées. Ensuite, seront présentées les différentes cartes graphiques (GPUs) utilisées pour la comparaison des temps de calculs obtenus. Pour finir, le modèle utilisé pour la DIV avec les premiers résultats obtenus seront présentés.

5.2 Matériels et méthodes

L'intérêt d'utiliser des données pertinentes pour satisfaire un modèle d'apprentissage correct a été montré dans le chapitre précédent. L'application étudiée étant un traitement

de radiothérapie externe, la balistique des traitements est régie par des lois physiques dont la cohérence peut être modélisée via des outils mathématiques tel que le machine-learning. L'ingrédient ajouté aux algorithmes de contrôles qualité pour effectuer la DIV est la distance radiologique. Elle apporte une information capitale sur les hétérogénéités traversées dans le patient par les rayons. Un grand nombre de distances radiologiques devront être calculées puisque pour chaque pixel déterminé, au moins une distance radiologique lui sera associée. Pour calculer les distances radiologiques massivement, le choix s'est porté sur l'utilisation de la bibliothèque CUDA permettant de faire du calcul parallèle intensif. De plus, dans cette section, est abordée l'introduction de ces distances radiologiques dans le modèle d'apprentissage en supplément de l'information tomodensitométrique CT.

5.2.1 Pertinence des distances radiologiques pour ce type de modèle

Dans le but de reconstruire une DIV, c'est à dire une DDA délivrée dans le patient, il est nécessaire d'utiliser un objet 3D modélisant le patient. L'imagerie 3D permettant de modéliser le patient pour la planification de traitement est le CT. Il possède l'information anatomique nécessaire au calcul de DDA planifiée du traitement. Le CT permet de manière fiable de modéliser la géométrie du patient.

Plaçons-nous dans la position d'un faisceau de rayons X partant de la tête de l'accélérateur de particules. Chaque rayon X va suivre un parcours différent suivant les lois de la physique sous-jacente, jusqu'à arrêter sa course lorsqu'il n'a plus d'énergie. Ces lois physiques d'atténuation du faisceau dépendent principalement de la distance parcourue et des sections efficaces des matériaux traversés. Ces informations sont disponibles dans la distance radiologique.

Dans le chapitre précédent, il a été montré que l'apprentissage dépendait des données d'entrées fournies. Il a été vu que si, il existait une cohérence entre chaque échantillon de données exposées, alors les algorithmes d'optimisation permettraient de définir un modèle représentatif de l'application souhaitée. Or, fournir l'information du CT, des pixels voisins du CT, la distance radiologique au niveau de l'EPID puis au niveau du CT ainsi que les distances radiologiques voisines apportent l'information principale pour la reconstruction de la DIV.

5.2.2 Modèle mathématiques utilisé

La distance radiologique correspond à la somme des distances traversées dans chaque pixel multiplié par leur densité associée. La formule mathématique associée au calcul de la distance radiologique est la suivante :

$$D_{rad} = \sum_i \sum_j \sum_k l(i,j,k)\rho(i,j,k)$$

avec $\{i,j,k\} \in N_r$ qui correspond au nombre total d'intersections traversées pas le rayon. De plus, $\rho(i,j,k)$ correspond à la valeur de densité du voxel (i,j,k) et l(i,j,k) à la

distance parcourue dans le voxel (i,j,k).

Le calcul rapide de la distance radiologique a été introduit par Siddon [203], puis suivi par Jacobs [204]. Ils ont montré que considérer le calcul sur chacun des voxels n'est pas judicieux, il est préférable de considérer les intersections parcourues sur chacune des dimensions. Ils ont aussi montré que dans le cas particulier du CT, chacune des dimensions étant orthogonales entres elles, chaque ensemble fournit des intersections parallèles et identiquement espacées. De ce fait, le pas dans les 3 dimensions est applicable récursivement. Considérons N le nombre de voxels de chaque dimension, la complexité du calcul passe de N^3 à $3N$, lui permettant d'être ainsi bien plus efficace.

Plutôt que d'utiliser les paramètres α tels que décrits par l'algorithme de Siddon [203] comme étant des valeurs paramétriques comprises entre 0 et 1, il a été choisi d'utiliser le calcul du vecteur directeur unitaire. Cela permet de déterminer la longueur de chaque pixel traversé plus rapidement. En effet, connaissant la position de la source du rayon et celle du voxel à considérer, il est possible d'en déduire le vecteur directeur unitaire avec la formule :

$$\overrightarrow{u_{AB}} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \frac{\overrightarrow{AB} \begin{pmatrix} x \\ y \\ z \end{pmatrix}}{\left\| \overrightarrow{AB} \right\|}$$

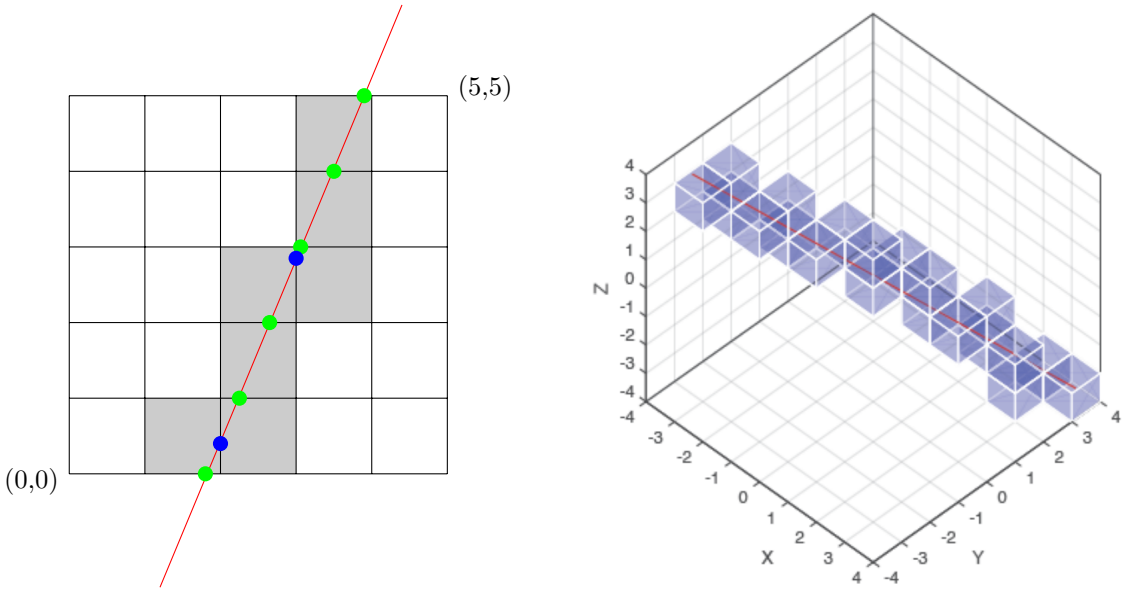
avec $\overrightarrow{AB}$ le vecteur de position du voxel (B) considéré en partant de la source (A).

De plus, à partir de la position de l'isocentre du patient, la position de chacune des intersections sur les trois dimensions est connue. Il est alors possible de calculer le module du vecteur AB croisant l'intersection considérée, anotée $\left\| \overrightarrow{AB} \right\|_{I^{i,j,k}_{x,y,z}}$ avec x,y,z correspondant aux trois dimensions et i,j,k aux intersections des voxels sur chacune des dimensions. De plus, on notera par la suite $I_{x_0,y_0,z_0,x_N,y_N,z_N}$ les intersections externes du CT et I_{x_1,y_1,z_1} les premières intersections réellement traversées par le rayon. Cette valeur s'obtient avec la formule :

$$\left\| \overrightarrow{AB} \right\|_{I^{i,j,k}_{x,y,z}} = \frac{\overrightarrow{AB}_{I^{i,j,k}_{x,y,z}}}{\overrightarrow{u_{AB}} \begin{pmatrix} x \\ y \\ z \end{pmatrix}}$$

Afin d'évaluer plus facilement le calcul effectué, une représentation sur deux dimensions d'une grille de 5*5 pixels est visualisable sur la Figure 5.1a. Il y apparait les points d'intersections des axes x et y, respectivement, bleus et verts. Une représentation en trois dimension est montrée en 5.1b.

Cette méthode consiste à calculer 3N modules. On verra dans la partie 5.2.3 qu'il est possible d'améliorer la complexité de l'algorithme en calculant, dans un premier temps, uniquement les modules de chaque intersection incluse dans l'ensemble des intersections traversées par le rayon.



(a) Grille avec un rayon traversant plusieurs intersections sur deux dimensions

(b) Grille avec un rayon traversant plusieurs intersections sur trois dimensions

FIGURE 5.1 – Grille avec un rayon traversant plusieurs intersections

Les modules à conserver sont régis par les formules :

$$d_{max} = min(\left\|\overrightarrow{AB}\right\|, max(\left\|\overrightarrow{AB}\right\|_{I_x}), max(\left\|\overrightarrow{AB}\right\|_{I_y}), max(\left\|\overrightarrow{AB}\right\|_{I_z}))$$

$$d_{min} = max(0, min(\left\|\overrightarrow{AB}\right\|_{I_x}), max(\left\|\overrightarrow{AB}\right\|_{I_y}), max(\left\|\overrightarrow{AB}\right\|_{I_z}))$$

Ces valeurs correspondent aux distances entre la source et le point d'entrée, respectivement, point de sortie dans le CT. Uniquement les valeurs entre ces deux bornes seront gardées. L'étape suivante consiste à fusionner l'ensemble des normes obtenues sur chacune des dimensions dans un ordre croissant régi par la formule :

$$\{\left\|\overrightarrow{AB}\right\|_I\} = \{d_{min}, fusion(\left\|\overrightarrow{AB}\right\|_{I_x}, \left\|\overrightarrow{AB}\right\|_{I_y}, \left\|\overrightarrow{AB}\right\|_{I_z}), d_{max}\}$$

Ayant l'ensemble des distances euclidiennes traversant chacune des dimensions dans un ordre croissant et connaissant la position du pixel (1,1,1) du CT, alors l'indice et la distance euclidienne exacte dans chacun des pixels traversé par le rayon sont accessibles.

L'ensemble des indices i,j,k sont calculables avec la formule :

$$\{i, j, k\} = (\{\left\|\overrightarrow{AB}\right\|_I\} * \overrightarrow{u_{AB}} \begin{pmatrix} x \\ y \\ z \end{pmatrix} - \overrightarrow{AB}_{I_{x_0,y_0,z_0}})/t(x, y, z)$$

avec $t(x, y, z)$ correspondant à la taille des voxels sur les trois dimensions.

De plus, il est possible d'obtenir la distance euclidienne traversée dans l'ensemble des voxels via la formule :

$$\{l(i,j,k)\} = \{\left\|\overrightarrow{AB}\right\|_I^o\} - \{\left\|\overrightarrow{AB}\right\|_I^{o-1}\}$$

avec $o \in \{1,..,N_r\}$.

Possédant l'ensemble des distances parcourues dans chacun des voxels et leur valeur de densité associée, il est alors possible de calculer la distance radiologique.

5.2.3 Méthodes algorithmiques utilisées

Le nombre de distances radiologiques à calculer est très important car la résolution de l'EPID et de la DDA à reconstruire est grande. Pour cette raison, l'algorithme utilisé doit être le plus efficace possible. De plus, l'extension de ces algorithmes sur une reconstruction de DDA 3D, augmenterait encore le nombre de distances radiologiquse à calculer. Il a été expliqué précédemment que la phase d'apprentissage peut posséder un grand nombre de données pour lesquelles la distance radiologique devra être associée. Puis pour finir, l'objectif en terme de temps de calcul pour la phase d'inférence, est d'être instantané.

Entrées : CT, *isocentre*
Sorties : D_r

$D_r \leftarrow 0$
Calcul de $\overrightarrow{u_{AB}}$
Calcul du module des intersections externes du CT en fonction de l'isocentre $\left\|\overrightarrow{AB}\right\|_{I_{x_0,y_0,z_0,x_N,y_N,z_N}}$
Calcul de d_{min} *et* d_{max}
Calcul de la distance vers l'intersection la plus proche de la source traversée par le rayon sur les trois dimensions $\left\|\overrightarrow{AB}\right\|_{I_{x_1,y_1,z_1}}$
Calcul des indices du voxel de la plus petite distance des trois dimensions
Placement de d_{min} *dans une variable temporaire*
tant que *Intersections comprisent entre* d_{min} *et* d_{max} *non traitées* **faire**
    Calcul de la distance entre la source et l'intersection suivante, traversée par le rayon $\left\|\overrightarrow{AB}\right\|_{I_{x,y,z}^{i,j,k}}$ *sur la dimension placée à l'itération précédente*
    Calcul de $l(i,j,k)$
    $D_r \leftarrow D_r + l(i,j,k) * CT(i,j,k)$
    Placement de la plus petite distance des trois dimensions dans la variable temporaire
    Incrémentation de l'indice de la dimension placée
fin

Algorithme 5.1 : Algorithme de calcul de distance radiologique

L'algorithme de calcul de distance radiologique est visible en 5.1. Dans cet algorithme, le calcul initial de d_{min} et d_{max} permet que seules les intersections réellement traversées par le rayon seront considérées, améliorant significativement l'accélération du calcul. Dans le cas du calcul de la distance radiologique où le rayon traverse diagonalement le CT, les 3N intersections devront être calculées. Cependant, si le rayon est orthogonal à deux des trois dimensions du CT, N intersections devront être calculées.

Les étapes de fusion, calcul des indices et de la distance traversée dans le voxel se font au cours de la même itération, permettant un gain de temps calculatoire conséquent.

Pour finir, le deuxième avantage d'effectuer ces 3 étapes dans la même itération concerne l'espace mémoire. En effet, il n'est pas nécessaire de stocker l'intégralité des informations des intersections. Seules la précédente et la suivante sont nécessaires pour le calcul de l(i,j,k).

5.2.4 CUDA pour l'efficacité du calcul numérique

Il a été évoqué précédemment le besoin de calculer de grandes quantités de distances radiologiques en un temps acceptable pour être utilisé en routine clinique. Ces distances radiologiques sont calculables de manière totalement indépendante. C'est pour cette raison que l'on s'est dirigé sur du calcul massivement parallèle.

C'est dans les années 2000 que les CPUs connaissent un effet de plafonnement dans leur évolution. Alors que jusque là, leurs capacités ne cessaient d'évoluer, leur structure ne leur permettait plus d'augmenter leur fréquence de fonctionnement.

Cependant, le potentiel de croissance pour la puissance du calcul parallèle réside actuellement, dans l'adjonction d'accélérateurs (GPUs) aux processeurs plutôt que d'utiliser des multi-processeurs CPUs. L'évolution dans ce domaine ne cesse d'augmenter ces dernières années.

L'évolution des (GPUs) ces trentes dernières années a été considérable, grâce notamment à l'apparition des jeux vidéos, où l'objectif était de toujours avoir des améliorations graphiques en 3D. Le marché du jeu vidéo étant très porteur, il a permis aux industriels de faire progresser le concept des (GPUs) initialement basé sur une approche vectorielle avant de prendre la direction d'une architecture massivement parallèle de type Single Instruction Multiple Threads (SIMT). Les capacités calculatoires de cette architecture se comptent en TeraFlops pour plusieurs milliers de coeurs de calculs alors que la bande passante mémoire se compte en centaines de Go/s.

Depuis 2007, il y a eu la révolution du General Purpose on GPU (GPGPU) avec la mise à disposition de CUDA pour les programmeurs. Ce framework permet alors un accès direct à l'architecture de la carte graphique (GPU) en utilisant le langage C ou C++.

Nvidia propose une large gamme de cartes graphiques (GeForce GTX et RTX, Quadro, NVS, Tesla, Titan et Tegra) et ont produit différentes architectures. La première génération de GPGPU constituée d'une architecture Tesla est la seule qui

continue à évoluer depuis, avec sa gamme de carte graphique (GPU). En parallèle, depuis 2010, succèdent chronologiquement les architectures Fermi, Kepler, Maxwell, Pascal, Volta, Turing et enfin Ampere. Chaque GPU possède une architecture dont les spécifications apparaissent dans le Tableau 5.1.

L'architecture des (GPUs) se décompose en 3 flots : Calculatoire, Mémoire et Instructions. Il suffit qu'un de ces 3 flots ne respecte pas ses capacités pour créer un goulot d'étranglement et effondrer les performances de la carte graphique (GPU).

Le flot calculatoire possède plusieurs composants hiérarchiquement structurés. Tout en haut de la pyramide se trouve le SoC du (GPU) contenant plusieurs clusters (GPCs). Chacun de ces GPC possèdent plusieurs Texture Processor Cluster (TPC) eux mêmes composés de deux Streaming Multiprocessor (SM). Un SM possède un grand nombre de coeurs. Il y a les coeurs CUDA pour le calcul simple précision des nombres entiers et flottants, les coeurs CUDA FP64 pour les calculs double précisions, les SFUs permettant de faire des opérations complexes comme la racine carré, le sinus ou cosinus, les coeurs tensor permettant le calcul matriciel, les coeurs ray-tracing permettant l'accélération du calcul virtuel de rayon à travers une scène (particulièrement utilisé pour le jeu vidéo).

Le (GPU) est un accélérateur, il doit être interfacé avec un CPU hôte. Cela implique que c'est le CPU qui déclenche l'execution de kernels. Le flot d'instructions se base sur plusieurs pipelines disponibles pour permettre au CPU de gérer les kernels ainsi que les unités de transferts mémoires au travers du bus PCIe. Au sein du (GPU), chaque GPC reçoit ses instructions à exécuter par kernel et aucune synchronisation entre clusters n'est possible. Cependant, au niveau des SMs, il existe des insructions de synchronisation. Chaque kernel lors de son appel par l'hôte, reçoit un nombre global d'instances à exécuter appelé grille. Cette dernière est composée de blocks correspondant aux entités à placer sur les GPCs. Ces blocks sont eux-mêmes composés de plusieurs threads représentant les entités à placer sur les SMs. Le découpage multi-hiérarchique des entités se fait de la manière suivante :

$$Taille\ de\ grille\ =\ Nombre\ de\ blocks\ \times\ Nombre\ de\ threads\ par\ block$$

Les instructions provenant d'un kernel sont placées dans le cache d'instruction de chaque SM. Ces instructions sont réparties dans les buffers d'instructions et les blocks de threads sont sous-découpés en warps de trente deux threads par l'ordonanceur de warps.

Le flot de mémoire est composé de plusieurs contrôleurs permettant d'intéragir avec la mémoire globale du (GPU) située à l'extérieur du SoC. Tout transfert de données entre l'espace mémoire du processeur hôte et la mémoire globale du (GPU) transite par le port PCIe puis le cache L2 puis par les contrôleurs mémoire afin d'atteindre la mémoire globale. Dans le cas d'utilisation de multiples (GPUs), le cheminement sera identique.

À plus bas niveau dans la hiérarchie du (GPU), se trouve au sein des SMs plusieurs branches permettant de transporter les données provenant de la mémoire globale jusqu'aux unités de calculs. La données transitent alors du cache L2 au cache L1, pour

	Fermi	Kepler	Maxwell	Pascal	Volta	Turing	Ampere
Fréquence Horloge (Ghz)	1,5	1,1	1,1	1,4	1,5	1,6	1,6
Nb transistors (milliards)	3	3,5	8,1	15,6	21,1	18,6	28,3
Nb TPCs	NA	NA	NA	30	42	36	41
Nb SM	16	8 SMX	24 SMM	60	84	72	82
Nb coeurs CUDA	512	1536	3072	3840	5376	4608	10496
Nb coeurs Tensor	NA	NA	NA	NA	612	576	328
Nb coeurs Ray-Tracing	NA	NA	NA	NA	NA	68	82
Fréquence horloge mémoire globale (GHz)	4	6	1,7	1,4	2	2	2,4
Bande passante DRAM (Gb/s)	192	192	336	750	900	672	936
DRAM Maximum (Go)	6	4	12	16	6	11	24
Nb SFU par SM	4	32	32	16	16	4	4
Nb LD/ST par SM	16	32	32	16	32	16	32
Nb ordonanceur warp	2	4	4	2	4	2	4

Tableau 5.1 – Spécifications des différentes architectures de GPUs Nvidia

ensuite atteindre les unités load/store (LS/ST).

Deux autres moyens d'accéder aux données présentes dans la mémoire globale sont possibles : La *constant* et la *texture memory* (fragment de la mémoire globale) uniquement accessible en lecture seule par le (GPU). La surface memory permet d'accéder au même espace mémoire que la texture memory en lecture et écriture.

Les variables scalaires initialisées lors de l'execution d'un kernel sur (GPU), sont stockées dans l'espace des registres propres au SM concerné. Sa vitesse d'accès est la plus rapide.

La mémoire locale est quant à elle un fragment de la mémoire globale dont l'utilisation repose sur un fonctionnment classique au moyen des caches L1 et L2. Sa portée est restreinte à chaque thread.

Pour finir, il existe la shared memory, elle est segmentée dans chaque SM (block) et présente une bande passante supérieure à celle de la mémoire globale.

Maintenant que les différents flots ont été abordés, il semble important de faire un point sur l'API CUDA. CUDA est à la fois un langage de programmation et l'API officiel de Nvidia. L'API est découpé en deux niveaux. Le premier correspond à CUDA Driver API, qui est une API C/C++ de bas niveau permettant un très haut niveau de contrôle des (GPUs) Nvidia à partir du processeur hôte.

Le second, concerne le CUDA Runtime API qui est une surcouche de la première, simplifiant grandement la gestion mémoire, la configuration et le lancement des threads sur le (GPU).

De plus, NVidia Cuda Compiler (NVCC) est le compilateur officiel pour le langage CUDA. Celui-ci repose sur la librairie NVidia Parallel Thread eXecution (NVPTX) développé par NVIDIA. NVCC permet de générer un code binaire exécutable sur (GPU) à partir d'un pseudo-code C ou C++.

Un kernel est une fonction particulière puisqu'elle peut être exécutée sur différents composants selon son type. Les trois types de kernels sont des fonctions classiques de C/C++ précédé du mot clé :

- ___global___ : appelé par le CPU mais exécuté par le (GPU)
- ___device___ : appelé et exécuté par le (GPU)
- ___host___ : appelé et exécuté par le CPU

L'appel d'un kernel de type ___global___, se fait avec la syntaxe suivante :

$kernel <<< nBlocks, nThreadPerBlocks >>> (arguments)$;

avec nBblocks et nThreadsBlocks de type dim3 représentant, respectivement la taille de la grille et la taille de chaque blocks. Cette instruction peut être suivie de cudaDeviceSynchronize() ; si une copie de la mémoire globale vers l'hôte doit être faite.

De plus, l'indice du thread de la grille exécuté par le (GPU) est accessible avec le kernel :

```
1 ___device___ int getId() {
2     int blockId = blockIdx.x + blockIdx.y * gridDim.x + gridDim.x * gridDim.y *
      blockIdx.z; int threadId = blockId * (blockDim.x * blockDim.y * blockDim.z)
      + (threadIdx.z * (blockDim.x * blockDim.y)) + (threadIdx.y * blockDim.x) +
      threadIdx.x;
3     return threadId;
4 }
```

Algorithme 5.2 : Algorithme d'obtention de l'indice du thread exécuté

Le transfert de données de l'hôte vers la mémoire globale se fait avec les étapes suivantes :

$cudaMalloc((void**)\&pCT, M * sizeof(float))$;

$cudaMemcpy(pCT, CT, M * sizeof(float), cudaMemcpyHostToDevice)$;

La première ligne correspond à l'allocation mémoire dans la mémoire globale. La seconde au transfert des données de l'hôte vers la mémoire globale.

Ensuite la récupération des résultats, donc le transfert des données de la mémoire globale vers l'hôte se fait avec la ligne :

$cudaMemcpy(result, pResult, M * sizeof(float), cudaMemcpyDeviceToHost)$;

Pour finir, ne pas oublier de désallouer la mémoire globale avec les lignes suivantes :

$cudaFree(pResult)$;

$cudaFree(pCT)$;

L'algorithme permettant de faire appel au kernel de l'agorithme parcouru en 5.2.3 est visible en 5.3.

Entrées : $CT,\ isocentre$
Sorties : $tabD_r$
1 $cudaMalloc((void**)\&pCT, M * sizeof(float))$;
  $cudaMalloc((void**)\&pResult, M * sizeof(float))$;
2 $cudaMemcpy(pCT, CT, M * sizeof(float), cudaMemcpyHostToDevice)$;
3 $CalculD_r <<< nBlocks, nThreadPerBlocks >>> (pResult, pCT, isocentre)$;
4 $cudaMemcpy(tabD_r, pResult, M * sizeof(float), cudaMemcpyDeviceToHost)$;
5 $cudaFree(pResult)$;
6 $cudaFree(pCT)$;

Algorithme 5.3 : Algorithme de calcul de distance radiologique

La compilation s'est faite par l'intermédiaire de l'outil NVCC. C'est une étape importante puisqu'elle permet de créer l'exécutable de l'application. Les options de compilation permettent d'améliorer l'efficacité de l'execution, notamment avec les

différents optimiseurs. De plus, l'utilisation de l'option c++11 a été choisie avec l'architecture de la carte. Cette dernière a permis d'utiliser des fonctions optimisées sur les itérateurs.

Par ailleurs, dans le cadre de la comparaison de l'efficacité de l'algorithme utilisé entre CUDA et le C++, il était important d'utiliser les mêmes options de compilation afin d'être le plus juste sur les résultats obtenus. L'option choisie a été l'option de débogage car les optimisations faites sur les deux langages comparés ne sont pas identiques.

Il a été évoqué un peu plus haut dans cette section, que l'ordonnanceur de warp allait placer les threads à exécuter dans des warps disponibles. Le nombre de threads dans un même block est limité au nombre de 1024. Il a par ailleurs été énoncé que ces threads sont dispatchés au nombre de 32 par warp, signifiant un nombre total de 32 warps par block.

Au sein du GPU, apparait un grand nombre de coeurs CUDA permettant de faire du calcul parallèle intensif. Le calcul de plusieurs distances radiologiques se fait de manière complétement indépendante et en ce sens, l'utilisation du GPU est pertinent. Cependant, l'information suivante est capitale, les threads lancés en parallèle dans un même warp exécutent les mêmes instructions imposées par le kernel. Autrement dit, si le calcul d'un thread dispose d'un plus grand nombre d'itérations, il mettra en attente l'ensemble des autres threads appartenant au même warp rendant le calcul moins efficace.

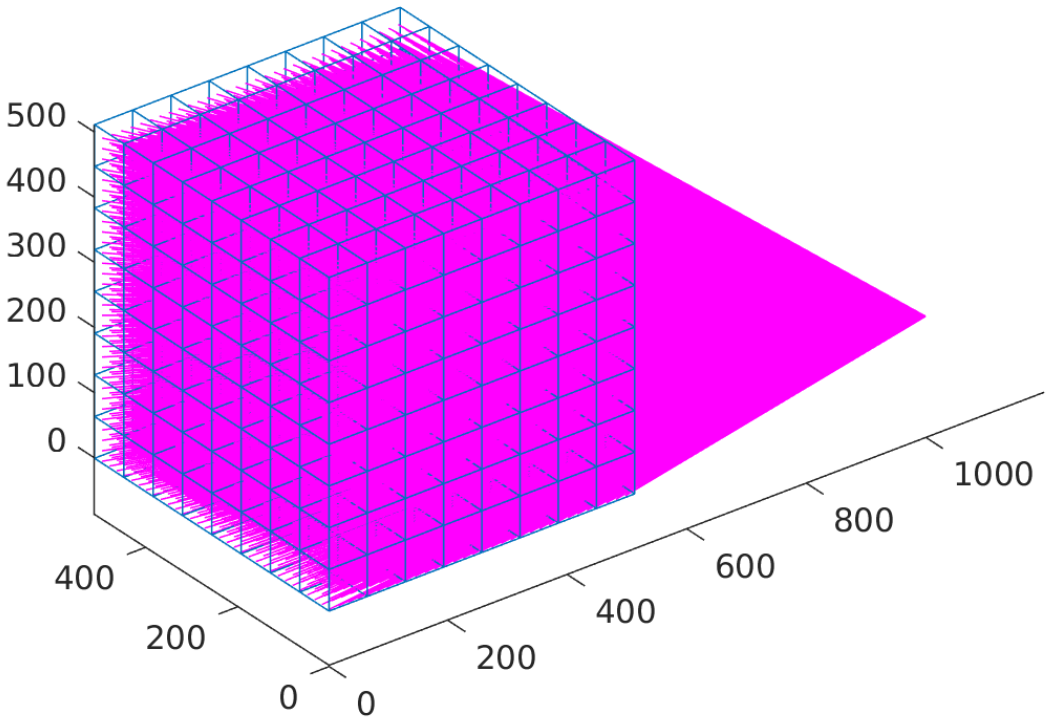


FIGURE 5.2 – Grille de calculs des distances radiologiques

C'est pourquoi, on peut visualiser sur la Figure 5.2, l'importance du placement des différents blocks de calculs. Chacun des cubes bleus représentés correspond à un block de calcul et possède 1024 threads. Chacun des threads correspond au calcul d'une distance radiologique. Il y a donc, dans un des cubes bleus 32 warps. Il est pertinent de placer les blocks de la sorte, car le nombre d'itérations considérées en fonction du nombre d'intersections traversées sera semblable pour des threads du même warp. Cependant, on imagine aisément que regrouper les 4 coins du CT dans un même warp rendrait le calcul totalement inefficace.

5.2.5 Tests sur différentes cartes (GPU)

L'application créée a été testée sur 3 cartes graphiques (GPUs) possédant chacune une architecture différente. La première d'entre elles, la moins puissante, est embarquée sur un ordinateur portable macbook pro mid 2014 avec un GPU Nvidia GeForce GTX 750M. Son architecture Kepler lui permet d'avoir deux SMX pour 384 coeurs CUDA. Sa fréquence d'horloge est de 941 MHz, avec un nombre de transistors de 1,27 milliards. Elle possède 2Go de DRAM avec une bande passante de 80 Go/s. Cette carte a principalement été utilisée pour le prototypage des algorithmes CUDA.

Le second GPU, est le modèle GeForce GTX 1080 embarqué sur un ordinateur de bureau possèdant le système d'exploitation Ubuntu 16.04. Cette carte possède une architecture Pascal qui lui permet d'avoir 20 TPCs, donc 40 SMs pour 2560 coeurs CUDA. Sa fréquence d'horloge est de 1,6 Ghz, avec un nombre de transistors qui s'élève à 7,2 milliards. Pour finir, elle possède 8Go de DRAM avec une bande passante de 320 Go/s.

La dernière carte graphique utilisée est une GeForce RTX 2080Ti embarquée dans une unité de calcul possèdant le système d'exploitation Ubuntu 18.04. Son architecture Turing lui fait monter un échelon puisqu'elle acquière 34 TPCs, donc 68 SMs pour 4352 coeurs CUDA. Sa fréquence d'horloge est 1,4 Ghz, avec 18,6 milliards de transistors. De plus, elle possède 11Go de DRAM avec une bande passante de 616Go/s.

Ces trois cartes ont pu être testées et comparées. En effet, elles ont permis d'évaluer les différences de performances obtenues sur un grand nombre de distances radiologiques calculées.

5.2.6 Encapsulation du code dans Matlab®

Les algorithmes de ces travaux ont été développés avec Matlab® comme il l'a été stipulé dans le Chapitre 4. Matlab® a permis un prototypage plus rapide, puisqu'un grand nombre de fonctionnalités sont déjà développées. Ces fonctionnalités ont principalement été utilisées pour le traitement d'images et pour l'utilisation des réseaux de neurones.

Ayant pour objectif un calcul de distance radiologique efficace, un interfaçage entre les programmes Matlab® et CUDA a été fait. Matlab® propose la mise en place de fonctions MEX (Matlab EXecutable) qui sont spécifiquement dédiées à l'encapsulation

de programmes C/C++. Ces fonctions sont précompilées à l'aide du compilateur mex (qui a besoin d'être paramétré avec gcc ou g++) de la même manière qu'on compile un programme C/C++. L'utilisation de ces fonctions a été expliquée en Annexe A. De plus, Matlab® a permis d'encapsuler les programmes CUDA avec le même principe et un fonctionnement identique via la création de fonctions MEXCUDA. Le compilateur mexcuda a préalablement besoin d'être configuré avec NVCC. Dès lors, le fichier binaire est créé et la fonction peut être utilisée dans Matlab®.

5.2.7 Prise en compte des hétérogénéités

Jusqu'à présent, le calcul de DDA se faisait, dans le cadre de la RCMI, comme montré au Chapitre 4, à partir d'un signal hétérogène dû à la position des lames dynamiques durant le traitement. Cependant, le milieu traversé par les rayons X était homogène (phase de contrôle qualité). Lorsque le patient est positionné sur la table durant le traitement, le milieu traversé devient hétérogène. Afin de prendre en considération cette hétérogénéité, le CT préalablement fait et ayant été contouré par les médecins est accessible. Il possède l'information géométrique du patient à partir des densités tissulaires qui le composent. Chacun des voxels est à prendre en considération.

Le CT est spécifique à chaque patient et permet d'obtenir un plan de traitement personnalisé. Énormément de paramètres sont à considérer d'où la difficulté de prédire une DIV spécifique. En effet, la forme des champs de traitements planifiés, les sections efficaces des milieux traversés, la profondeur à considérer, la balistique du traitement, la physique sous-jacente, l'énergie du faisceau et bien d'autres grandeurs influencent la DIV.

Cependant, l'espoir est dans le traitement d'un grand spectre de cas lors de la phase d'apprentissage afin de reconstruire une DIV au plus juste. Il est important de comprendre que pour un voxel considéré, plusieurs chemins peuvent amener à un même résultat, l'application est surjective. Et ce n'est pas parce que tous les patients sont différemment composés qu'il n'existe pas de lien sur la reconstruction de DIV. Il est essentiel pour ce type d'application de satisfaire un grand nombre de cas obtenus par l'intermédiaire d'une grande base de données afin d'acquérir un modèle qui sera représentatif de l'application.

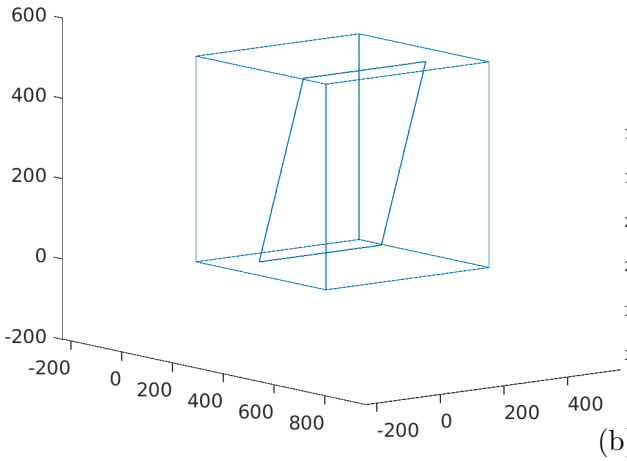
5.2.8 Utilisation des données pertinentes

Le choix des données utilisées pour la phase d'apprentissage est capital. Le prochain objectif est de reconstruire une DDA *in-vivo* 2D à partir de l'image EPID, le CT et les distances radiologiques sous différents points de vue. L'image EPID est celle récupérée post-traitement des patients. L'étude a été faite pour deux types de traitements, chacun ayant des énergies différentes afin de parcourir un plus grand nombre de cas. Les traitements sont la radiothérapie conformationnelle avec une énergie de traitement de 6 MeV et la radiothérapie conformationnelle à modulation d'intensité avec une énergie de 25 MeV. Ces types de traitements ont déjà été étudiés dans le cadre du CQ, beaucoup

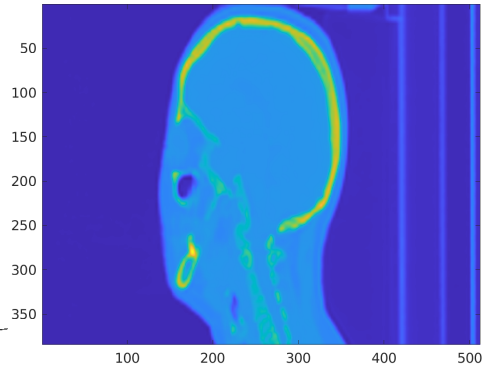
de caractéristiques seront réutilisées pour la DIV.

La base de données utilisée pour la DIV possède 6, respectivement, 8 ensembles EPIDs/TPSs pour la RC et la RCMI. Tous correspondent à des localisations encéphaliques totales, respectivement, prostatiques. Les traitements ont été effectués avec des accélérateurs Varian®. Toutes les images EPIDs obtenues possèdent 384 × 512 pixels. Les images ont été acquises en mode intégré avec une DSD de 150 cm. Elles ont toutes été corrigées du FF et du DF dont l'explication est donnée en 3.3.3. Un filtre seuil (*threshold*) a été appliqué sur les images EPIDs permettant de définir le masque à appliquer sur l'ensemble des données fournies à l'algorithme d'apprentissage.

La seconde donnée d'entrée pertinente correspond au CT du patient. Le CT est une image 3D et doit être traitée afin d'obtenir une map 2D échantillonnée sur les mêmes coordonnées que la DDA planifiée (TPS). Sur la Figure 5.3a, apparait le plan 2D d'un



(a) Position du plan 2D du CT à récupérer



(b) Plan 2D du CT obtenu avec un angle de 160°

FIGURE 5.3 – Position et visualisation du CT 2D associé

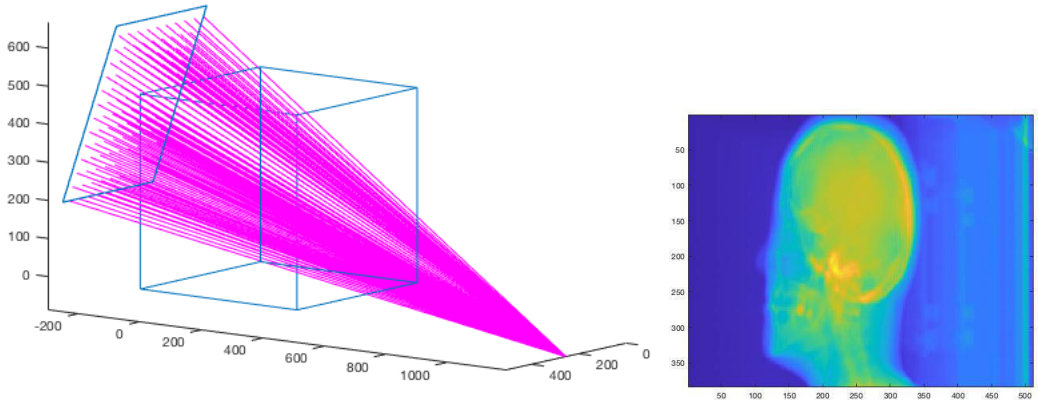
CT à extraire. Ce plan 2D doit respecter certaines hypothèses :

— La première, correspond au plan qui doit être orthogonal au rayon considérant l'isocentre du patient. Cela permettra d'avoir le plan 2D du CT et le plan de l'EPID parallèle entre eux.

— La deuxième concerne la prise en compte de l'élargissement du faisceau avec la profondeur. Il est alors nécessaire d'échantillonner la map 2D récupérée afin d'avoir la même résolution que l'image EPID. Sur la Figure 5.3b est montré un exemple de CT 2D pouvant être extrait pour l'algorithme d'apprentissage automatique.

Une troisième donnée consiste à apporter l'information des distances radiologiques projetées sur l'EPID. La Figure 5.4a montre que pour chaque position de pixels de l'EPID la distance radiologique est calculée. Cette valeur permet d'avoir une information sur la densité totale qu'a pu traverser le rayon dans le patient.

De plus, sur la Figure 5.4b apparait une map 2D de distances radiologiques projetées au niveau de l'EPID à partir du même CT vu en 5.3b. On remarque que l'information



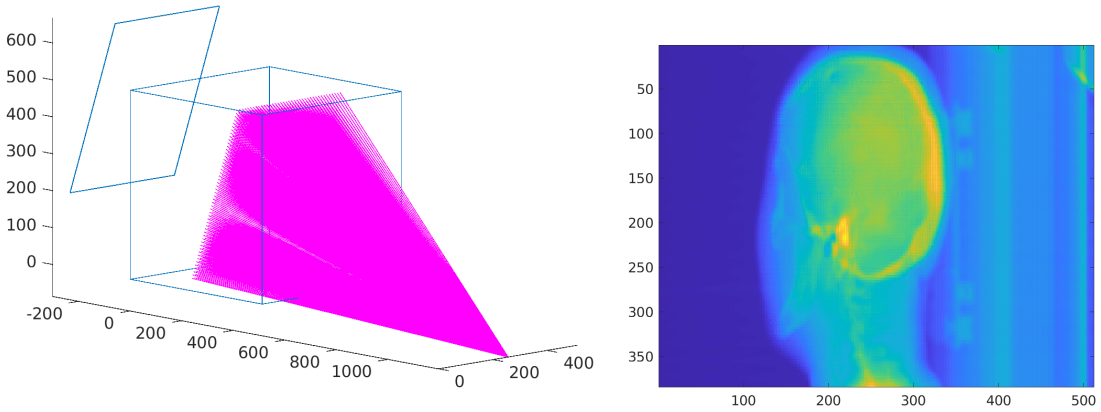
(a) Distances radiologiques projetées sur l'EPID

(b) Image des distances radiologiques obtenu avec un angle de 160°

FIGURE 5.4 – Position et visualisation des distances radiologiques

obtenue diffère complélement puisque sur l'une d'entre elles, l'information d'une coupe est extraite, alors que sur l'autre, une mémorisation de toutes les densités traversées est gardée.

Pour finir, une quatrième donnée est considérée, il s'agit de la map 2D des distances radiologiques projetées sur la coupe du CT orthogonale au rayon considérant l'isocentre du patient comme vu sur la Figure 5.3a. Sur la Figure 5.5a sont visibles les différentes distances radiologiques prises en compte. On remarque que leur course s'arrête au niveau du plan CT 2D désiré.



(a) Distances radiologiques projetées dans le CT sur le plan parallèle de l'EPID

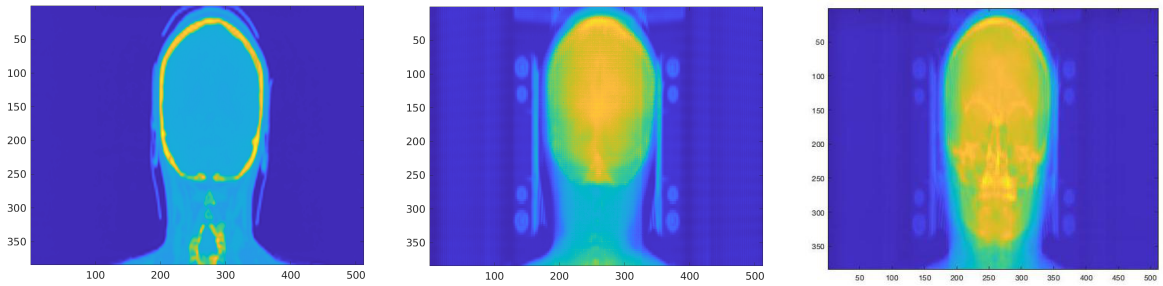
(b) Image des distances radiologiques obtenues avec un angle de 160°

FIGURE 5.5 – Position et visualisation des distances radiologiques sur le plan isocentrique du CT

Sur la Figure 5.5b apparait une map 2D de distances radiologiques projetées au niveau de la coupe du même CT vu en 5.3b. On remarque, à nouveau que l'information obtenue diffère puisque sur la première, c'est toujours l'information d'une coupe qui est extraite, alors que sur l'autre, une mémorisation des densités traversées jusqu'à

la coupe orthogonale au rayon considérant l'isocentre du patient est gardée. De plus, la différence obtenue avec la Figure 5.4b peut être remarquée puisque, l'information contenue dans la partie avant du visage du patient apparait dans ce dernier.

La complémentarité des trois dernières données exposées précédemment, est encore plus flagrante sur la Figure 5.6. A Gauche, sur la Figure 5.6a, est visualisable la coupe 2D du CT qui est orthogonale au rayon considérant l'isocentre du patient. Au milieu, sur la Figure 5.6b, est montrée l'image des distances radiologiques projetées sur la coupe 2D CT. Enfin, à droite, la Figure 5.6c, donne l'image des distances radiologiques projetées au niveau de l'EPID. Ces trois images sont complétement différentes et chacune d'elle apporte une information essentielle du patient. La première détermine la densité de la coupe considérée pour la DDA. La deuxième, apporte la mémorisation des densités traversées jusqu'à la coupe considérée pour la DDA. La troisième possède la mémorisation des densités traversées jusqu'à l'EPID.



(a) Coupe du CT 2D parallèle à l'EPID

(b) Distances radiologiques projetées sur la coupe CT

(c) Distances radiologiques projetées sur l'EPID

FIGURE 5.6 – Images prises avec angle de 180°

5.2.9 Dosimétrie *in-vivo*

L'objectif final de ces travaux était de réussir à prédire la DIV à partir de l'imageur portal EPID. Pour ce faire, comme vu dans le paragraphe précédent, à l'information en provenance de l'EPID, ont été ajoutées les distances radiologiques sous différentes formes et telles que présentées sur la Figure 5.7.

Revenons sur les données d'entrées réellement fournies, ainsi que sur le modèle établi pour prédire une DIV. Tout comme pour le modèle établi dans la prédiction du CQ, le choix de considérer chacun des pixels en tant qu'échantillon de donnée a été fait. Pour chacune des données présentées en 5.2.8, le choix d'utiliser les informations voisines a été établi de sorte à considérer le même nombre d'entrées pour les 4 métadonnées d'entrées. Cela permettait de pondérer de manière équitable chacune des métadonnées dans le modèle.

De plus, la normalisation a été établie comme pour le modèle de contrôle qualité. Les 4 métadonnées possèdent la même résolution (384×512 valeurs). L'image EPID a subi une inversion de signal, les valeurs des pixels de l'image EPID ont été multipliées par le

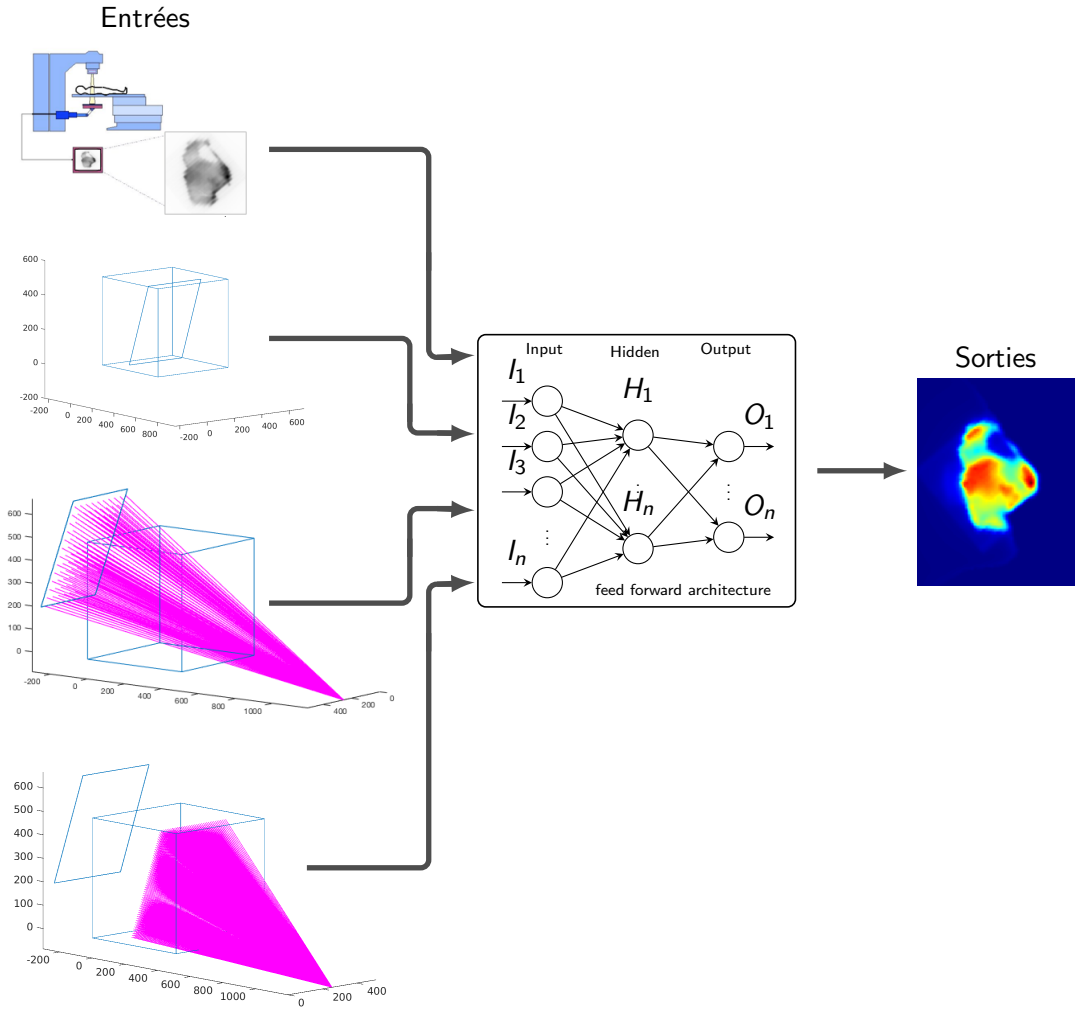


FIGURE 5.7 – Schéma explicatif des données d'entrées/sorties pour la DIV

nombre de trames, un *threshold* de 15%, respectivement, 10% pour la RC et la RCMI. Ce *threshold* a permis d'établir un masque qui a été appliqué sur les 4 métadonnées d'entrées, ainsi que la donnée de sortie qui a été préalablement multipliée par le facteur « DoseGridScaling ». Ceci, a permis de considérer le même nombre d'échantillons pour chacune des métadonnées.

L'architecture du modèle choisi dans ce chapitre est le FFNN. Ce modèle basique a été choisi plutôt que le CNN car le framework utilisé de Matlab® ne permet pas de faire un apprentissage multimodal à partir de CNN. Cependant, on a vu précédemment, que l'architecture feed-forward était efficace. Le nombre d'entrées du modèle est de 488 valeurs, respectivement, 902 valeurs pour la RC et la RCMI. Le nombre de neurones de la couche cachée est de 732, respectivement, 1353. La couche de sortie possède 1 neurone. L'algorithme d'optimisation utilisé durant la phase d'apprentissage est la méthode des gradients conjugués. Les fonctions d'activations utilisées sont la fonction sigmoïd pour

la couche cachée et la fonction linéaire pour la couche de sortie. L'initialisation des poids est faite de manière aléatoire et la fonction coût à optimiser est la MSE.

5.3 Résultats et discussions

Dans la partie précédente, les méthodes de calculs utilisées pour obtenir les distances radiologiques de manière efficace ont été abordées. Enfin, l'architecture et les paramètres du modèle utilisé ont été énoncés.

Dans cette partie, une comparaison de performances obtenues pour le calcul de distances radiologiques entre les différentes cartes graphiques (GPUs) utilisées a été faite. Cette comparaison a été complétée avec le calcul sur CPU à partir des langages C++ et Matlab®. Ensuite, une analyse des résultats obtenus sur la prédiction des DDAs *in-vivo* a été produite avant de finir sur les perpectives à envisager pour la suite de ces travaux.

5.3.1 Évaluation des performances obtenues avec CUDA

Le calcul des distances radiologiques est un processus qui est par nature, lent. En effet, la résolution du CT étant élevée pour un besoin de plus grandes précisions sur la composition du patient, n'aidant en rien l'efficacité de ce calcul.

Cependant, utilisant des algorithmes de réseaux de neurones, il est important d'utiliser des données qui modélisent précisément le patient pour une prédiction de DDA fiable. Ce calcul est lent car pour une distance radiologique il est nécessaire de calculer la distance parcourue dans chacun des voxels traversés dans le CT.

L'ambition étant de calculer une DIV en temps-réel, il était nécessaire de traiter les données en temps-réel afin de les fournir au modèle de prédiction. La méthode la plus efficace pour obtenir un grand nombre de distances radiologiques est le calcul massivement parallèle comme présenté en 5.2.4. Trois cartes graphiques GPUs ont été présentées. Elles possèdent chacune une architecture différente.

Afin d'évaluer les performances de calculs obtenues sur ces trois cartes graphiques GPUs, nous avons opté pour le calcul d'un maximum de distances radiologiques en un temps de 25 secondes et sous certaines conditions. L'EPID a été considéré avec sa résolution la plus grande, donc 768×1024 pixels et le CT avec $512 \times 512 \times 512$ voxels. Le calcul de distances radiologiques projetées sur l'EPID étant le plus lent des calculs, il a été choisi de considérer uniquement ces derniers. Ce qui a varié, c'est le nombre d'angles de traitement considérés sur 360° comme montré en Figure 5.8.

Le GPU le moins performant est le GeForce GTX 750M. En 25 secondes, il a permis de calculer 25 angles de traitements. Son nombre de distances radiologiques obtenues s'élève à 19 660 800.

La deuxième carte graphique (GPU) testée est la GeForce GTX 1080. Cette fois-ci,

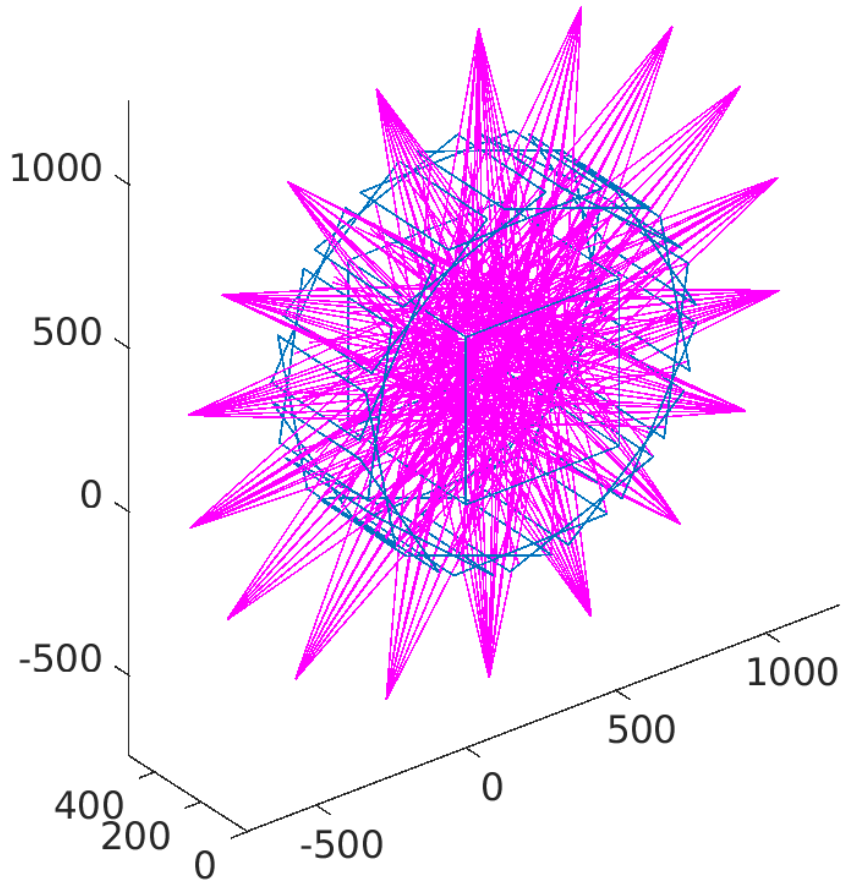


FIGURE 5.8 – Schéma pour la comparaison du temps de calcul des distances radiologiques

en 25 secondes, ont été obtenus 520 angles de traitements. Le nombre de distances radiologiques calculées est 408 944 640. Cette carte graphique (GPU) a calculé avec un facteur 20,8 fois plus rapidement que la précédente.

Pour finir, la carte graphique (GPU) la plus performante des trois testées, correspond à la GeForce RTX 2080Ti. Elle a permis de résoudre 2600 angles de traitements portant son nombre de distances radiologiques obtenues à 2 044 723 200. Ainsi, elle a été 104 fois plus rapide que le GPU GeForce GTX 750M.

5.3.2 Comparaison de l'efficacité entre CUDA, C++ et Matlab®

Afin de voir l'ampleur de l'efficacité des algorithmes utilisés sur un GPU, il était important de la comparer au temps de calcul obtenu à partir d'un CPU. Le processeur utilisé est un Intel Core i7 cadencé à 2,5GHz. Pour se faire, le programme CUDA a été comverti en langage C++ et compilé avec les mêmes options que le programme CUDA tel qu'exposé en 5.2.4.

Afin d'évaluer la différence de performance, la même méthode qu'appliquée en 5.3.1 a été utilisée. Le nombre d'angle obtenus en 25 secondes s'élève au nombre de 3 pour

FIGURE 5.9 – Image EPID d'un traitement encéphalique total sans patient

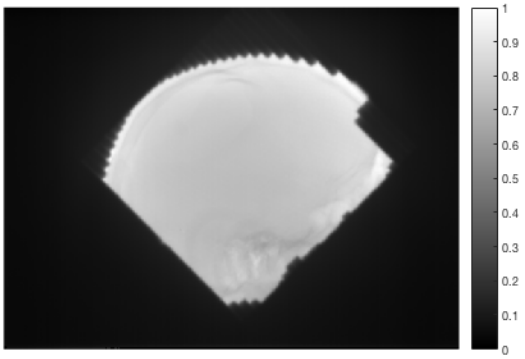


FIGURE 5.10 – Image EPID d'un traitement encéphalique total avec patient

un total de distances radiologiques calculées de 2 359 296. Si l'on compare le nombre de valeurs de distances radiologiques obtenues entre le GPU GeForce RTX 2080Ti et ce processeur i7, on obtient un facteur d'efficacité de 866,67 pour les mêmes algorithmes utilisés.

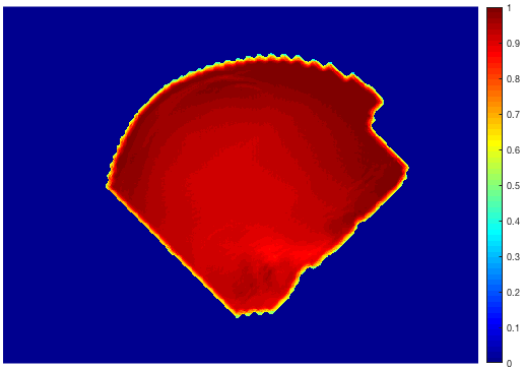
Pour finir et compléter la comparaison, l'algorithme a été implémenté sur Matlab®. Le processeur utilisé pour l'exécution est le Intel Core i7 et il a permis d'obtenir l'équivalent de 0,4 angle soit 314 572 distances radiologiques. En comparant ce résultat, avec celui de la plus performante des cartes graphiques (GPUs) utilisée dans ces travaux, on obtient un facteur de 6375.

5.3.3 Évaluations des résultats de calcul de dosimétrie *in-vivo*

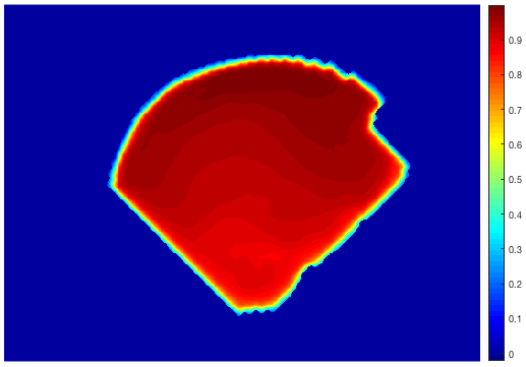
La prédiction de DDA *in-vivo* a été établie pour deux types de traitements à deux énergies différentes. Tous deux ont été effectués à partir de l'accélérateur de particules Varian®. Dans un premier temps, c'est le traitement de radiothérapie conformationnelle à modulation d'intensité qui sera abordé. Ce traitement a été effectué avec une énergie de 6 MeV. Pour rappel, ce type de traitement utilise le bras de l'accélérateur et le collimateur multi-lames à position fixe durant l'irradiation. Ceci entraîne un champ d'irradiation homogène justifié par la Figure 5.9, qui a été prise sans patient durant le CQ. Cependant, lorsque l'irradiation s'effectue avec la présence du patient, on peut apercevoir sur la Figure 5.10, l'apparition d'un signal un peu plus hétérogène dû à la géométrie du patient. Les deux mesures nous montrent la différence de signal avec et sans patient pour un même traitement. La mesure de l'EPID avec patient a été faite dans les conditions énoncées en 2.3.1 et en 5.2.8. L'irradiation s'est faite avec une DSD de 150 cm. La phase d'apprentissage a été effectuée à partir d"une base de données impliquant 6 ensembles d'entrées/sorties.

Sur la Figure 5.11a apparait la DDA prédite par le modèle de réseau de neurones et sur la Figure 5.11b la DDA originalement planifiée. Le traitement considéré durant la phase d'inférence, concerne la phase de traitement d'un encéphale total entrainant

de grands champs d'irradiation. On remarque visuellement quelques différences entre les deux distributions notamment sur les bords de champs. On remarque aussi qu'aux endroits où les hétérogénéités tissulaires sont présentes, la DDA prédite semble être en accord avec celle planifiée. Cependant, un $\gamma_{index}(3\%,\ 3\ mm)$ global de 76.36% a été obtenu, ne montrant pas que les algorithmes étaient capables de reconstruire une DDA *in-vivo* de manière précise.



(a) Distribution de dose absorbée *in-vivo* prédite en radiothérapie conformationnelle



(b) Distribution de dose absorbée *in-vivo* planifiée en radiothérapie conformationnelle

FIGURE 5.11 – Comparaison entre image calculée et planifiée pour la RC en DIV

La carte des γ est visualisable sur la Figure 5.12. Cette carte permet de visualiser spatialement les valeurs obtenues et surtout celles qui ne respectent pas le critère $\gamma_{index}(3\%,\ 3\ mm)$. On remarque que le nombre de points qui ne respectent pas le critère est bien plus élevé que dans le cas du CQ. La plupart des points qui ne respectent pas le critère sont placés en bord de champs. On remarque aussi sur l'EPID que certains bords de champs sont placés dans le vide, laissant penser à un petit décalage du patient.

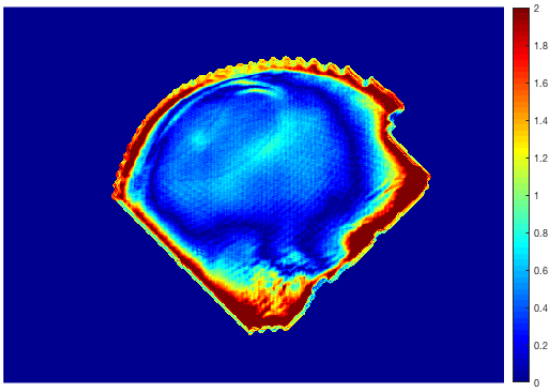


FIGURE 5.12 – Carte des $\gamma_{index}(3\%,\ 3\ mm)$ entre l'image prédite et planifiée du calcul de DIV en CRT

Le deuxième traitement concerne la RCMI. Ce traitement a été effectué avec une énergie de 25 MeV. Pour rappel, ce type de traitement utilise le bras de l'accélérateur fixe et le collimateur multi-lames dynamique durant l'irradiation. Ceci entraîne un

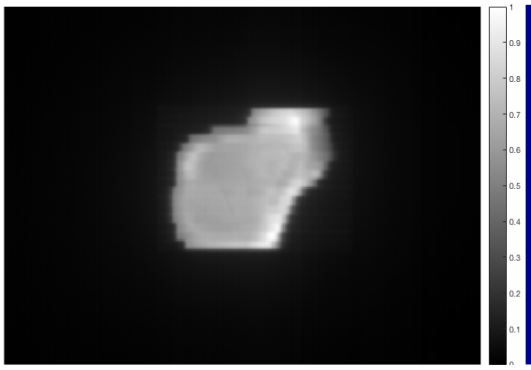


FIGURE 5.13 – Image EPID d'un traitement prostatique avec patient

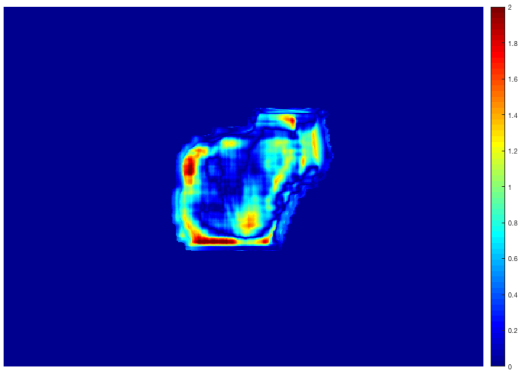
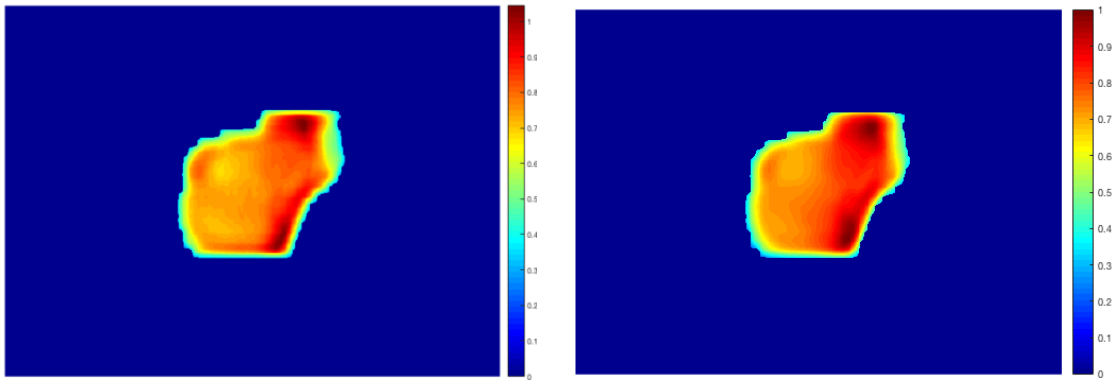


FIGURE 5.14 – Carte des $\gamma_{index}(3\%,\ 3\ mm)$ entre l'image prédite et planifiée du calcul de DIV en RCMI

champ d'irradiation possédant un signal hétérogène, ajouté à la présence du patient comme le montre la Figure 5.13. La mesure de l'EPID avec patient a été faite dans les conditions énoncées en 2.3.2 et en 5.2.8. L'irradiation s'est faite avec une DSD de 150 cm. De plus, la phase d'apprentissage a été effectuée à partir d'une base de données impliquant 8 ensembles d'entrées/sorties.

Sur la Figure 5.15a apparait la DDA prédite par le modèle de réseau de neurones et sur la Figure 5.15b la DDA originalement planifiée. Le traitement considéré durant la phase d'inférence, concerne la phase de traitement d'une prostate. On visualise à nouveau quelques différences entre les deux distributions principalement en bord de champs. De plus, un $\gamma_{index}(3\%,\ 3\ mm)$ global de 86.78% a été obtenu, se rapprochant plus vers un résultat souhaité, cependant, on ne peut conclure sur la capacité des algorithmes à reconstruire une DDA *in-vivo* de manière précise.



(a) Distribution de dose absorbée *in-vivo* prédite en radiothérapie conformationnelle à modulation d'intensité

(b) distribution de dose absorbée *in-vivo* planifiée en radiothérapie conformationnelle à modulation d'intensité

FIGURE 5.15 – Comparaison entre image calculée et planifiée pour la RCMI en DIV

La carte des γ est affichée sur la Figure 5.14. Cette carte permet de visualiser spatia-

lement les valeurs obtenues puis celles qui ne respectent pas le critère $\gamma_{index}(3\%,\ 3\ mm)$. Le nombre de points qui respectent le critère est plus élevé que pour la RC. Cependant, la grande majorité des points qui ne respectent pas le critère sont placés en bord de champs.

5.4 Pour aller plus loin

Dans cette partie, les points limitant les résultats obtenus dans la section 5.3.3 seront abordés avec des potentiels axes d'amélioration. De plus, sera évoquée l'extension des algorithmes vers le 3D à partir de différentes modélisations possibles. Pour commencer, on a pu voir précédemment sur deux types de traitements différents à deux énergies distinctes que les résultats obtenus n'étaient pas ceux attendus. Afin d'analyser la cause de ces résultats et de les comprendre, plusieurs aspects sont à prendre en considération.

Le premier, concerne le nombre de patients considérés lors de la phase d'apprentissage. En effet, lors du CQ, une petite base de données patient pouvait suffire car il y avait une relation directe possible à établir entre les entrées et sorties. Cependant, pour la DIV, ce n'est plus le cas. D'autres entrées, telles que la géométrie du patient doivent être prises en considération restreignant grandement la généralisation de notre modèle car chacun des patients a une géométrie qui lui est propre. D'où l'importance de considérer un grand nombre de patients dans la base de données d'apprentissage. Pour faire référence à la Figure 4.3b, considérer la géométrie du patient revient à agrandir l'espace dimensionnel du modèle et offre plus d'incertitudes pour les échantillons non traités lors de la phase d'apprentissage.

Le deuxième porte sur la multimodalité des données à considérer. En effet, le modèle établi lors de la phase de contrôle qualité peut être associé à la fonctionnalité monomodale puisque toute l'information d'entrée provenait de l'imageur portal EPID. Cependant, les modèles construits pour la DIV font appel à des entrées contenant des informations différentes telles que l'EPID et le CT. La multimodalité augmente la complexité du modèle car il faut que les données d'entrées restent cohérentes entre elles avec pour objectif d'avoir un lien mathématique avec la sortie. C'est pourquoi, les échelles prises et l'information qu'elles apportent ont toute leur importance. De plus, l'apport équilibré d'information provenant des modalités dans le modèle doit être géré. En effet, il faut garder en tête que la mesure apportant une information sur le traitement délivré est l'EPID. Le CT apporte l'information de la géométrie obtenue prétraitement. Il ne faut pas que l'information provenant du CT soit prise en priorité sur l'EPID dans le modèle, risquant de perdre l'information de la mesure de l'irradiation délivrée.

Le troisième porte sur l'information transmise lors de la phase d'apprentissage. Il est nécessaire que la sortie reflète réellement la vérité puisque dans le cas contraire, le modèle apprendra de mauvaises informations. On rappelle que lors de la phase de contrôle qualité, l'irradiation se fait sans rien sur la table minimisant la possibilité d'avoir une mauvaise mesure EPID. Cependant, lorsque le patient est positionné sur

la table, une mauvaise position ou un amaigrissement entre séances de traitement pourrait venir nuire à la vérité terrain. Pour cela, utiliser l'information du CBCT à la place du CT porterait tout son sens. En outre, les données d'entrées et de sorties ont été choisies par des physiciens médicaux sur des traitements qui leurs semblaient correctement appliqués. Afin d'éviter ce genre de perturbations, il existe des fantômes anthropomorphes constitués avec des densités tissulaires identiques aux êtres humains. Utiliser des fantômes anthropomorphes pour la phase d'apprentissage permettrait de supprimer les risques de mouvements ou d'évolutions corporelles.

Le quatrième se situe au niveau du modèle d'apprentissage établi. Les limitations du frameworks Matlab® ont été un handicap pour pouvoir construire des modèles plus évolués. En effet, la toolbox Deep Learning ne permet pas de construire des modèles complexes avec des métadonnées de dimensions différentes. De plus, elle ne permet pas de considérer des modèles multimodaux à partir de modèles plus complexes, ce qui a favorisé l'utilisation du modèle feed-forward pour la DIV. L'utilisation de map 2D des distances radiologiques en entrée du modèle n'a pas permis la modélisation de la diffusion en profondeur. Or, considérer les pixels voisins sur trois dimensions plutôt que sur deux dimensions aurait été un réel atout pour la modélisation de la diffusion. Convertir les algorithmes établis sur Matlab® vers un framework possédant une communauté active en IA tel que PyTorch ou Tensorflow serait un réel atout.

Le cinquième se situe sur la précision des calculs de doses absorbées planifiés. On a vu en 4.2.3, que dans certaines conditions le TPS ne calculait pas une DDA correcte, pouvant contraindre le modèle à apprendre de mauvaises informations. Utiliser durant la phase d'apprentissage, des DDAs calculées à partir d'algorithmes Monte-Carlo pourrait apporter une plus grande précision. Cependant, gardons en tête que l'objectif reste de comparer la DDA prédite par le modèle avec celle planifiée par le TPS.

Pour finir, durant ces travaux, seuls des modèles 2D ont été établis. Concernant la DIV, considérer des modèles 3D pourrait donner de meilleurs résultats puisqu'ils permettraient d'avoir l'information de la dimension de profondeur pour mieux modéliser la diffusion. Cependant, l'information de l'EPID étant en 2D, deux possibilités seront envisageables pour obtenir une information 3D. La première consiste à répéter le signal 2D sur une matrice 3D en tenant compte de la divergence du faisceau et de la profondeur afin d'obtenir un signal 3D.

La deuxième solution concerne le traitement VMAT abordé en 2.3.3 et en 3.3.2. Il existe un mode d'acquisition continu de l'image EPID qui consiste à sauvegarder le signal à différents angles d'irradiation. Tout comme, pour un CT, il est possible de reconstruire un objet 3D à partir de plans 2D sous différents angles rétroprojetés, il est envisageable de reconstruire un signal EPID 3D à partir des projections 2D provenant de l'acquisition continue. Enormément de méthodes de reconstruction existent dans la littérature, classées en deux catégories : les méthodes analytiques et les méthodes itératives. Les premières sont reconnues pour être efficaces et approximatives, alors que les secondes sont précises et lentes. Un essai de reconstruction itératif a été établi à partir de la méthode OSEM (Ordered Subset Expectation Maximisation). Ce calcul a permis d'entrevoir la difficulté de réutiliser le signal obtenu. En effet, les images obtenues à ces

énergies (de l'ordre du MeV) sont très bruitées dû à un effet Compton prépondérant. De plus, après analyse des images reconstruites, seuls les traitements offrant de grands champs durant les deux arcs d'irradiation, apportaient une réelle information sur les hétérogénéités traversées. Cependant, dans la majeure partie des cas, les traitements VMAT entrainent de petits champs d'irradiation. La problématique dans ce cadre, est que l'absence d'information sur une projection entraine la suppression de signal dans les autres projections même si ces dernières possèdent des petits champs d'irradiation. Autrement dit, la reconstruction avec les méthodes actuelles n'apportent pas autant d'informations qu'espéré. Néanmoins, des méthodes de reconstruction partielles ne supprimant pas la possibilité d'avoir une absence de signal sont à l'étude.

5.5 Conclusion

Pour conclure, durant ce chapitre ont été abordées les différentes métaentrées utilisées pour le modèle d'apprentissage, à commencer, par le calcul des distances radiologiques. Une méthodologie algorithmique a permis d'en obtenir un grand nombre avec des temps d'ecécution compatibles avec des objectifs de temps-réel. Ensuite, plusieurs formes de ces distances radiologiques ont été associées à des informations provenant de l'EPID et du CT pour construire le modèle d'apprentissage.

Le modèle d'apprentissage utilisé a fourni des résultats de prédiction de DIV pour deux types de traitement à deux énergies différentes. Ces résultats n'ont pas été à la hauteur de ceux obtenus durant la phase de contrôle qualité. De tels résultats étaient attendus car la complexité de modéliser la géométrie du patient dans ces modèles d'apprentissage est grande. La fin de ce chapitre a été consacrée à quelques axes d'améliorations pouvant amener à avoir de meilleurs résultats.

Conclusion générale

L'utilisation de techniques d'apprentissage automatique pour la prédiction de DDA est un défi pour les prochaines années. Beaucoup de recherches ont été établies à partir de méthodes analytiques donnant de bons résultats de manière ciblée sur certains équipements. L'avantage principal de l'utilisation de techniques d'apprentissage automatique est dans la généralisation des résultats sur plusieurs équipements et différents types de traitement. Cela est faisable car le modèle se construit en fonction des données qui lui sont transmises. Trouver l'algorithme qui s'applique dans plusieurs cas de fonctionnement permet de le généraliser sur différents équipements. La condition est d'avoir un signal en entrée des algorithmes qui soit répétable et reproductible.

Pour arriver à cet objectif, il a été présenté dans la première partie, un ensemble de techniques d'apprentissage automatique présent dans la littérature. Un sous ensemble de ces techniques ont été réutilisées dans l'application des travaux. La seconde partie porte sur l'application qu'est la radiothérapie externe et son fonctionnement. Elle nécessite des procédures de traitements bien spécifiques afin d'obtenir un traitement précis et modulé pour chaque patient. La troisième partie présente l'utilisation et l'acquisition de l'imageur portal EPID pour la dosimétrie. Différentes méthodes ont été abordées dans la littérature, dans ces travaux, une nouvelle approche est proposée. De plus, le traitement des images EPIDs par plusieurs corrections a été montré.

C'est dans cette quatrième partie, que la contribution de ces travaux a réellement commencée à être abordée avec l'explication des modèles d'apprentissage utilisés. Le choix d'un ordre précis de contribution a été appliqué. L'idée était de définir un cadre simple en premier lieu afin de déterminer le modèle en ayant le minimum de paramètres non contrôlables. Par la suite, un ensemble de rajouts de paramètres a permis de complexifier le modèle pour obtenir un résultat prometteur.

Le choix s'est porté sur l'utilisation de FFNNs appliqués au traitement de radiothérapie conformationnelle sur la phase de contrôle qualité à partir de l'accélérateur linéaire de particules Varian® pour une énergie de 6 MeV. Un grand nombre de paramètres et d'hyper-paramètres ont été fixés en fonction des données transmises au modèle. Le modèle établi a permis d'avoir de très bons résultats avoisinant un $\gamma_{index}(2\%,\ 2\ mm)$ global de 99,8% montrant la capacité de ce premier algorithme à reconstruire une DDA. Les algorithmes ont ensuite été étendus vers la phase de contrôle qualité pour le traitement de radiothérapie conformationnelle à modulation d'intensité. Les résultats obtenus dans ce cadre ont été corrects puisque le $\gamma_{index}(2\%,\ 2\ mm)$ global associé était de 99,7%. Durant cette partie, on a pu apprécier la capacité des algorithmes à s'adapter en fonction des données transmises. Ensuite, l'extension des algorithmes s'est portée vers de nouvelles énergies de traitement. Les entrées fournies au modèle d'apprentissage ont été adaptées en rapport à la physique sous-jacente à ce type de traitement. Il a été montré que ces nouvelles adaptations algorithmiques ont permis d'obtenir un $\gamma_{index}(2\%,\ 2\ mm)$ global de 99,3%.

L'étape suivante a conduit à entreprendre les calculs à partir d'images provenant d'un accélérateur linéaire de particules Elekta®. Les deux types de traitement précédemment étudiés ont été abordés. De nouvelles procédures d'acquisition des données EPIDs et un nouveau système de planification de traitement (TPS) n'ont pas empêché les algorithmes d'apprentissage d'être efficaces à la fois pour la RC et la RCMI.

De plus, pour valider le calcul effectué par les algorithmes, une défaillance visible sur l'image EPID a été simulée afin d'entrevoir la réaction obtenue sur la prédiction. Cette défaillance est une mauvaise position de lame durant le traitement. Il a été noté que la prédiction a considéré cette dernière de manière cohérente malgré qu'aucune mauvaise position de lame n'ait été présente dans la base de données d'apprentissage.

Pour finir avec la phase de contrôle qualité, des modèles plus complexes ont été implémentés utilisant des CNNs. Ils ont été testés sur plusieurs types de traitements ainsi que différentes énergies. Les résultats obtenus sont comparables au modèle précédent, cependant, on remarque une meilleure modélisation de la diffusion.

Le cinquième et dernier chapitre porte sur l'évolution des algorithmes vers la prédiction de dose absorbée *in-vivo*. Pour reconstruire cette prédiction, le choix d'utiliser le CT sous différentes formes en entrée de l'algorithme a été fait. Une information particulière a été choisie, il s'agit de la distance radiologique. Cette distance mémorise l'information du parcours traversé par le rayon. Elle est par nature, lente à obtenir puisque l'accumulation de chaque voxel traversé doit être calculée. Des algorithmes utilisant du calcul massivement parallèle ont permis d'obtenir ces distances en un temps compatible avec un objectif d'obtention en temps réel. Le modèle d'apprentissage s'est effectué à partir d'un FFNN. Ses entrées correspondent au CT, à l'EPID, aux distances radiologiques projetées sur l'EPID et sur le CT. Ce modèle a été appliqué sur la RC et la RCMI, chacune avec une énergie de traitement différente. Les résultats obtenus n'ont pas été convaincants avec des scores de $\gamma_{index}(3\%,\ 3\ mm)$ global qui s'élèvent à 76,36% pour la RC et 86,78% pour la RCMI.

Les perspectives de ces travaux sont nombreuses tant cette nouvelle approche tient ses promesses. Encore énormément de travail reste à accomplir par la suite avant de pouvoir utiliser ce type d'algorithmes en routine clinique. Une des perspectives serait de convertir tous les algorithmes en Python afin d'utiliser des frameworks dont la communauté IA est grandissante tels que PyTorch ou Tensorflow. La construction de graphes dynamiques (c'est à dire des entrées de tailles évolutives) et complexes sont accessibles dans ce type de frameworks augmentant considérablement le champs des possibles. De plus, les modèles 3D sont accessibles avec des temps de calcul optimisés pour des structures conséquentes.

Par la suite, le cheminement logique serait de trouver un modèle plus efficient pour la DIV 2D et de montrer leurs efficacité sur différents accélérateurs de particules. Ce modèle devrait pouvoir tenir compte de maps 3D des distances radiologiques afin d'apporter l'influence que peut contenir la profondeur sur la diffusion. De plus, dans l'objectif d'améliorer la phase d'apprentissage, il serait important de considérer une bien plus grande base de données. Reconstruire une DIV à partir de modèles d'apprentissage

reste un challenge considérable tant le nombre de paramètres à tenir compte est grand. Ensuite, il faudrait étendre les algorithmes vers un modèle 3D de prédiction. Considérer une nouvelle fois les distances radiologiques permettrait de modéliser la profondeur pour chacune des couches de la troisième dimension. Cette prédiction 3D pourrait être appliquée à la fois au CQ et à la DIV.

La finalité de ces travaux serait d'étendre les algorithmes vers des traitements VMAT. Ce sont des traitements qui sont de plus en plus utilisés en clinique et dont la DIV reste complexe à déterminer.

Annexe A

Fiches techniques fonctions MEX

A.1 Description de l'objectif

Le principal objectif est de pouvoir interfacer Matlab avec une bibliothèque externe codée en C/C++. Pour réussir à l'atteindre, l'idée est d'utiliser un fichier MEX contenant une fonction MEX (signifiant Matlab EXecutable) qui permet l'intéraction entre Matlab et le code C/C++.

La même procédure est utilisable pour encapsuler un programme CUDA, dans ce cas, c'est un fichier MEXCUDA qu'il faut compiler.

A.2 Forme du fichier MEX

Fichier Mex

```
#include <mex.h>
#include <stdio.h>
#include <stdlib.h>
#include "Ajouter.h"
void mexFunction( int nlhs,mxArray *plhs[],int nrhs,const mxArray *prhs[]){
   double z=0. ;
   z=Ajouter(mxGetScalar(prhs[0]),mxGetScalar(prhs[1])) ;
   plhs[0] = mxCreateDoubleScalar(z) ;
   printf("Le résultat de l'operation suivante est :
%lf + %lf = %lf",mxGetScalar(prhs[0]),mxGetScalar(prhs[1]),z) ;
}
```

A.3 Compilation du fichier MEX

Afin de pouvoir utiliser la fonction, il faut d'abord la compiler. En effet, contrairement à Matlab, les langages C et C++ ne sont pas des langages interprétés. Pour compiler avec Mex, nous allons utiliser sensiblement la même syntaxe que lors de l'utilisation de GCC. La commande, ressemblera à la suivante :

mex Mex_nom_de_fonction.extension ' − I/chemin/des/include'

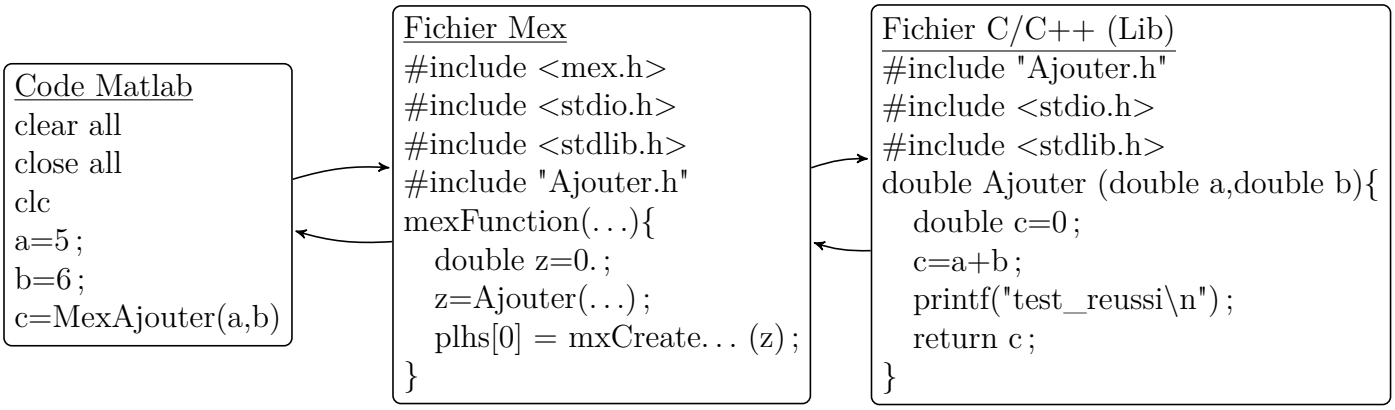
Lors de l'utilisation d'une librairie extérieure, la commande devient :

mex Mex_nom_de_fonction.extension ' − L/chemin/des/bibliothèques'
nom_de_bibliothèque.extension

Une fois correctement compilée, la fonction est utilisable sur Matlab (sur console, dans un script ...).

A.4 Utilisation de bibliothèques extérieures

Comme nous l'avons stipulé auparavant, l'idée est de réutiliser un code C/C++ déjà compilé en tant que bibliothèque. Voici la structure complète qu'aura notre programme :



A.5 Problèmes de compatibilités rencontrées

Le fait de vouloir utiliser des librairies extérieures déjà compilées peut poser des problèmes de compatibilté avec Matlab. En effet, il est nécessaire que la librairie soit utilisable par la version Matlab.

Pour cela voici un récapitulatif des utilisations compatibles.

A.5.1 Windows 32 bits

— Compilation du Mex-File sans librairies extérieures en C et C++

— Compilation du Mex-File sans librairies extérieures mais en utilisant des fonctions en C et C++

— Compilation du Mex-file avec librairies statiques extérieures en utilisant l'IDE Visual Studio C++ et en compilant avec Visual Studio C++ uniquement pour des librairies en C

— Compilation du Mex-file avec librairies statiques extérieures en utilisant l'IDE CodeBlocks et en compilant avec Visual Studio C++ uniquement pour des librairies en C
— Compilation du Mex-file avec librairies statiques extérieures en utilisant l'IDE CodBlocks et en compilant avec Visual Studio C++ uniquement pour des librairies en C

A.5.2 Windows 64 bits

Pour pouvoir compiler des bibliothèques extérieures compatibles avec une version Matlab Windows 64 bits, il faut faire de la cross-compilation sous linux. Pour cela, il est possible d'utiliser une machine virtuelle et d'installer les paquets mingw32 et mingw-w64 avec les commandes *sudo apt-get install mingw32* et *sudo apt-get install mingw-w64*. En parallèle, il faudra installer mingw64-TDM sur Windows et modifier le fichier matlab mexopts.bat afin d'utiliser gnumex pour compiler.

— Compilation du Mex-File sans librairies extérieures en C et C++
— Compilation du Mex-File sans librairies extérieures mais en utilisant des fonctions en C et C++
— Compilation du Mex-file avec librairies dynamiques extérieures en utilisant la cross-compilation sous Linux avec les commandes standards de compilation avec *x86_64-w64-mingw32-* devant chaque commande. Ceci fonctionne pour des librairies en C et C++

A.5.3 Linux

— Compilation du Mex-File sans librairies extérieures en C et C++
— Compilation du Mex-File sans librairies extérieures mais en utilisant des fonctions en C et C++
— Compilation du Mex-file avec librairies statiques extérieures en compilant avec le compilateur GCC. Ceci fonctionne pour des librairies en C et C++

Remarque 1 : Pour compiler une librairie statique, il faut utiliser les commandes *g++ -c -fPIC fichier.cpp* et *ar -q -s libnom.a fichier.o* Pour compiler une librairie dynamique par la cross-compilation, il faut utiliser les commandes *x86_64-w64-mingw32-g++ -c -fPIC fichier.cpp* et *x86_64-w64-mingw32-g++ -shared -o libnom.so fichier.o*

Remarque 2 : Toutes ces utilisations permettent la mémorisation des variables globales utilisées dans les bibliothèques.

Remarque 3 : L'utilisation de mexcuda fonctionne de la même manière que la fonction mex. Le compilateur à utiliser s'appelle mexcuda.

Table des figures

1.1 Schéma d'ensemble de l'IA . . . 6
1.2 Descriptif des phases de l'IA . . . 6
1.3 Schéma de fonctionnement de la régularisation Dropout . . . 8
1.4 Frise chronologique des principaux auteurs d'article de ce chapitre . . . 13
1.5 Modèle du perceptron . . . 15
1.6 Modèle du HN . . . 17
1.7 Représentation d'un Auto-encodeur . . . 18
1.8 Modèle du RNN . . . 19
1.9 Modèle du GAN . . . 24
1.10 Neurone biologique . . . 28
1.11 Neurone artificiel . . . 29
1.12 Architecture du réseau de neurones feed-forward . . . 31
1.13 Architecture classique d'un CNN . . . 33
1.14 Représentation du produit de convolution entre deux matrices . . . 34
1.15 Représentation de l'opération max-*Pooling* . . . 35
1.16 Structure d'un réseau de neurones . . . 37

2.1 Imageries utilisées pour le diagnostic . . . 48
2.2 Contourages effectués par le médecin . . . 49
2.3 Phénomènes physiques pour l'imagerie . . . 50
2.4 Procédure de planification . . . 52
2.5 Vision intérieure de la tête et de la section accélératrice du LINAC . . . 57
2.6 Image représentative du MLC . . . 58
2.7 Technique d'irradiation en VMAT avec plusieurs points d'entrées . . . 60

3.1 Représentation des principaux points de mesures pour la DIV [121] . . . 62
3.2 LINAC . . . 65
3.3 Composition physique de l'EPID [131] . . . 65
3.4 Collimateur multi-lames . . . 66
3.5 Images de corrections apportées au signal brut de l'EPID . . . 70
3.6 Réponse en dose absorbée linéaire à partir d'une mesure EPID . . . 71

4.1 Représentation de la position des images acquises lors du contrôle qualité 79
4.2 Schéma explicatif des données d'entrées/sorties pour les réseaux de neurones classiques . . . 85
4.3 Exemple de régression linéaire . . . 87
4.4 Équilibre électronique . . . 88
4.5 Schéma explicatif des données d'entrées/sorties pour les CNNs . . . 90

4.6 Architecture du CNN 91
4.7 Exposition du γ_{index} dans un cas typique 93
4.8 Comparaison entre image EPID originale et simulée pour la RCMI 6 MeV Varian® à partir d'un RNA classique 94
4.9 Image EPID pour la RC 6 MeV Varian® à partir d'un RNA classique . 95
4.10 Comparaison entre image calculée et planifiée pour la RC 6 MeV Varian® à partir d'un RNA classique 96
4.11 Image EPID pour la RCMI 6 MeV Varian® à partir d'un RNA classique 97
4.12 Comparaison entre image calculée et planifiée pour la RCMI 6 MeV Varian® à partir d'un RNA classique 97
4.13 Régression des différents ensembles de données de la phase d'apprentissage 99
4.14 Performance et régression de l'apprentissage pour la RCMI 6 MeV Varian® à partir d'un RNA classique 100
4.15 Image EPID pour la RCMI 6 MeV Varian® à partir d'un RNA classique 100
4.16 Comparaison entre image calculée et planifiée pour la RCMI 6 MeV Varian® à partir d'un RNA classique 100
4.17 Carte des $\gamma_{index}(2\%,\ 2\ mm)$ pour la RCMI 6 MeV Varian® à partir d'un RNA classique 101
4.18 Simulation d'une mauvaise position de lame pour la RCMI 6 MeV Varian® à partir d'un RNA classique 101
4.19 Image EPID pour la RCMI 25 MeV Varian®à partir d'un RNA classique 102
4.20 Comparaison entre image calculée et planifiée pour la RCMI 25 MeV Varian® à partir d'un RNA classique 103
4.21 Régression de la phase d'apprentissagepour la RC 6 MeV Elekta® à partir d'un RNA classique 104
4.22 Image EPID pour la RC 6 MeV Elekta® à partir d'un RNA classique . 104
4.23 Comparaison entre image calculée et planifiée pour la RC 6 MeV Elekta® à partir d'un RNA classique 105
4.24 Régression de la phase d'apprentissage pour la RCMI 6 MeV Elekta® à partir d'un RNA classique 106
4.25 Image EPID pour la RCMI 6 MeV Elekta® à partir d'un RNA classique 106
4.26 Comparaison entre image calculée et planifiée pour la RCMI 6 MeV Elekta® à partir d'un RNA classique 106
4.27 Performance et régression de l'apprentissage pour la RCMI 6 MeV Varian® à partir d'un CNN 107
4.28 Image EPID pour la RCMI 6 MeV Varian® à partir d'un CNN 108
4.29 Comparaison entre image calculée et planifiée pour la RCMI 6 MeV Varian® à partir d'un CNN 108
4.30 Carte des $\gamma_{index}(2\%,\ 2\ mm)$ pour la RCMI 6 MeV Varian® à partir d'un CNN 109
4.31 Simulation d'une mauvaise position de lame pour la RCMI 6 MeV Varian® à partir d'un CNN 109
4.32 Image EPID pour la RCMI 25 MeV Varian® à partir d'un CNN 110
4.33 Comparaison entre image calculée et planifiée pour la RCMI 25 MeV Varian® à partir d'un CNN 110

4.34 Carte des $\gamma_{index}(2\%,\ 2\ mm)$ pour la RCMI 25 MeV Varian® à partir d'un CNN 111
4.35 Simulation d'une mauvaise position de lame pour la RCMI 25 MeV Varian® à partir d'un CNN 111

5.1 Grille avec un rayon traversant plusieurs intersections 117
5.2 Grille de calculs des distances radiologiques 124
5.3 Position et visualisation du CT 2D associé 127
5.4 Position et visualisation des distances radiologiques 128
5.5 Position et visualisation des distances radiologiques sur le plan isocentrique du CT 128
5.6 Images prises avec angle de 180° 129
5.7 Schéma explicatif des données d'entrées/sorties pour la DIV 130
5.8 Schéma pour la comparaison du temps de calcul des distances radiologiques 132
5.9 Image EPID d'un traitement encéphalique total sans patient 133
5.10 Image EPID d'un traitement encéphalique total avec patient 133
5.11 Comparaison entre image calculée et planifiée pour la RC en DIV . . . 134
5.12 Carte des $\gamma_{index}(3\%,\ 3\ mm)$ entre l'image prédite et planifiée du calcul de DIV en CRT 134
5.13 Image EPID d'un traitement prostatique avec patient 135
5.14 Carte des $\gamma_{index}(3\%,\ 3\ mm)$ entre l'image prédite et planifiée du calcul de DIV en RCMI 135
5.15 Comparaison entre image calculée et planifiée pour la RCMI en DIV . . 135

Liste des tableaux

1.1 Nombre de poids intervenant sur le réseau 35

4.1 Résultats obtenus pour les différents traitements 112

5.1 Spécifications des différentes architectures de GPUs Nvidia 121

Liste des Algorithmes

1.1 Algorithme d'apprentissage 41

5.1 Algorithme de calcul de distance radiologique 118
5.2 Algorithme d'obtention de l'indice du thread exécuté 123
5.3 Algorithme de calcul de distance radiologique 123

Bibliographie

[1] Y. LeCun, Y. Bengio & G. Hinton, "Deep Learning", *Nature*, vol. 521, pp. 436–44, May 2015.

[2] E. J. Topol, "High-performance medicine: the convergence of human and artificial intelligence", *Nature Medicine*, vol. 25, no. 1, pp. 44–56, 2019.

[3] P. Meyer, V. Noblet, C. Mazzara & A. Lallement, "Survey on deep learning for radiotherapy", *Comput. Biol. Med.*, vol. 98, pp. 126–146, Jul. 2018.

[4] S. Chen, S. Zhou, J. Zhang, F.-F. Yin, L. B. Marks & S. K. Das, "A neural network model to predict lung radiation-induced pneumonitis", *Medical physics*, vol. 34, pp. 3420–3427, Sep. 2007.

[5] R. Güntürkün, "Determining the Amount of Anesthetic Medicine to Be Applied by Using Elman's Recurrent Neural Networks via Resilient Back Propagation", *Journal of Medical Systems*, vol. 34, no. 4, pp. 493–497, Aug. 2010.

[6] X. Zhu, Y. Ge, T. Li, D. Thongphiew, F.-F. Yin & J. Q. Wu, "A planning quality evaluation tool for prostate adaptive IMRT based on machine learning", *Medical physics*, vol. 38, pp. 719–726, Feb. 2011.

[7] M. Sauget, « Parallélisation de problèmes d'apprentissage par des réseaux neuronaux artificiels. Application en radiothérapie externe », Theses, Université de Franche-Comté, déc. 2007.

[8] A. Vasseur, « Réseaux de neurones artificiels pour la dosimétrie des faisceaux de photons en radiothérapie externe. Étude de faisabilité », Theses, Université de Franche-Comté, oct. 2009.

[9] K. Coeytaux, E. Bey, D. Christensen, E. S. Glassman, B. Murdock & C. Doucet, "Reported radiation overexposure accidents worldwide, 1980-2013: a systematic review", *PloS one*, vol. 10, no. 3, e0118709–e0118709, Mar. 2015.

[10] M. Herman, J Kruse & C Hagness, "Guide to clinical use of electronic portal imaging", *Journal of applied clinical medical physics / American College of Medical Physics*, vol. 1, pp. 38–57, Feb. 2000.

[11] K. Langmack, "Portal Imaging", *The British journal of radiology*, vol. 74, pp. 789–804, Oct. 2001.

[12] M. Herman, J. Balter, D. Jaffray, *et al.*, "Clinical use of electronic portal imaging: Report of AAPM Radiation Therapy Committee Task Group 58", *Medical physics*, vol. 28, pp. 712–37, Jun. 2001.

[13] M. G. Herman, "Clinical Use of Electronic Portal Imaging", *Seminars in Radiation Oncology*, vol. 15, no. 3, pp. 157 –167, 2005.

[14] W. van Elmpt, L. McDermott, S. Nijsten, M. Wendling, P. Lambin & B. Mijnheer, "A literature review of electronic portal imaging for radiotherapy dosimetry", *Radiotherapy and oncology : journal of the European Society for Therapeutic Radiotherapy and Oncology*, vol. 88, pp. 289–309, Sep. 2008.

[15] B. Mijnheer, S. Beddar, J. Izewska & C. Reft, "In vivo dosimetry in external beam radiotherapy", *Medical physics*, vol. 40, p. 070 903, Jul. 2013.

[16] H. Zou & T. Hastie, "Regularization and variable selection via the elastic net", *Journal of the Royal Statistical Society Series B*, vol. 67, pp. 768–768, Feb. 2005.

[17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever & R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[18] S. Ioffe & C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", in *Proceedings of the 32Nd International Conference on International Conference on Machine Learning*, ser. ICML'15, vol. 37, Lille, France: JMLR.org, 2015, pp. 448–456.

[19] K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space", *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901. eprint: https://doi.org/10.1080/14786440109462720.

[20] W. S. McCulloch & W. Pitts, "A logical calculus of the ideas immanent in nervous activity", *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[21] D. O. Hebb, *The organization of behavior: A neuropsychological theory*. New York: Wiley, Jun. 1949.

[22] A. M. TURING, "I.—COMPUTING MACHINERY AND INTELLIGENCE", *Mind*, vol. LIX, no. 236, pp. 433–460, Oct. 1950. eprint: http://oup.prod.sis.lan/mind/article-pdf/LIX/236/433/9866119/433.pdf.

[23] Newell, Shaw & Simon, "Report on a general problem-solving program, RAND Corp", *Memo P-1584 (30 December 1958)*, 1958.

[24] Rosenblatt, *The perceptron, a perceiving and recognizing automaton, Project Para*. Cornell Aeronautical Laboratory, 1957.

[25] F. Rosenblatt, "The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain", *Psychological Review*, pp. 65–386, 1958.

[26] H. J. Kelley, "Gradient Theory of Optimal Flight Paths", *ARS Journal*, vol. 30, no. 10, pp. 947–954, 1960.

[27] McCarthy, "Recursive Functions of Symbolic Expressions and Their Computation by Machine, Part I", *Communications of the ACM*, vol. 3, no. 4, pp. 184–195, 1960.

[28] T. Cover & P. Hart, "Nearest neighbor pattern classification", vol. 13, pp. 21–27, 1967.

[29] M. Minsky & S. Papert, *Perceptrons: An Introduction to Computational Geometry*. Cambridge, MA, USA: MIT Press, 1969.

[30] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, 1982. eprint: https://www.pnas.org/content/79/8/2554.full.pdf.

[31] T. Kohonen, "Self-organized formation of topologically correct feature maps", *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.

[32] D. E. Rumelhart, G. E. Hinton & R. J. Williams, "Learning representations by back-propagating errors", *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[33] G. E. Hinton & T. Sejnowski, "Learning and relearning in Boltzmann machines", in *Parallel distributed processing: Explorations in the microstructure of cognition*, Cambridge, MA: MIT Press, 1986, pp. 282–317–.

[34] P. Smolensky, "Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1", in, D. E. Rumelhart, J. L. McClelland & C. PDP Research Group, Eds., Cambridge, MA, USA: MIT Press, 1986, ch. Information Processing in Dynamical Systems: Foundations of Harmony Theory, pp. 194–281.

[35] D. Broomhead & D. Lowe, "Radial basis functions, multi-variable functional interpolation and adaptive networks", *ROYAL SIGNALS AND RADAR ESTABLISHMENT MALVERN (UNITED KINGDOM)*, vol. RSRE-MEMO-4148, Mar. 1988.

[36] H. Bourlard & Y. Kamp, "Auto-association by multilayer perceptrons and singular value decomposition", *Biological Cybernetics*, vol. 59, pp. 291–294, 1988.

[37] T. Kohonen, "An introduction to neural computing", *Neural Networks*, vol. 1, no. 1, pp. 3 –16, 1988.

[38] LeCun, Boser, Denker, *et al.*, "Backpropagation applied to handwritten zip code recognition", *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[39] C. Watkins, "Learning From Delayed Rewards", PhD thesis, King's College, Cambridge, 1989.

[40] J. L. Elman, "Finding structure in time", *Cognitive Science*, vol. 14, no. 2, pp. 179 –211, 1990.

[41] C. Cortes & V. Vapnik, "Support-Vector Networks", *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[42] Tin Kam Ho, "Random decision forests", in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, 1995, 278–282 vol.1.

[43] S. Hochreiter & J. Schmidhuber, "Long Short-term Memory", *Neural computation*, vol. 9, pp. 1735–80, Dec. 1997.

[44] M. Schuster & K. K. Paliwal, "Bidirectional recurrent neural networks", *IEEE Trans. Signal Processing*, vol. 45, pp. 2673–2681, 1997.

[45] Y. LECUN, L. BOTTOU, Y BENGIO & P. HAFFNER, "Gradient-Based Learning Applied to Document Recognition", *Proceedings of the IEEE*, vol. 86, pp. 2278 –2324, Dec. 1998.

[46] W. MAASS, T. NATSCHLÄGER & H. MARKRAM, "Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations", *Neural Computation*, vol. 14, no. 11, pp. 2531–2560, 2002. eprint: https://doi.org/10.1162/089976602760407955.

[47] H. JAEGER & H. HAAS, "Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication", *Science*, vol. 304, no. 5667, pp. 78–80, 2004. eprint: https://science.sciencemag.org/content/304/5667/78.full.pdf.

[48] G.-B. HUANG, Q.-Y. ZHU & C.-K. SIEW, "Extreme learning machine: Theory and applications", *Neurocomputing*, vol. 70, no. 1, pp. 489 –501, 2006, Neural Networks.

[49] M. RANZATO, C. POULTNEY, S. CHOPRA & Y. LECUN, "Efficient Learning of Sparse Representations with an Energy-based Model", in *Proceedings of the 19th International Conference on Neural Information Processing Systems*, ser. NIPS'06, Canada: MIT Press, 2006, pp. 1137–1144.

[50] Y. BENGIO, P. LAMBLIN, D. POPOVICI & H. LAROCHELLE, "Greedy Layer-Wise Training of Deep Networks", in *Advances in Neural Information Processing Systems 19*, B. SCHÖLKOPF, J. C. PLATT & T. HOFFMAN, Eds., MIT Press, 2007, pp. 153–160.

[51] G. E. HINTON, "Training Products of Experts by Minimizing Contrastive Divergence", *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002. eprint: https://doi.org/10.1162/089976602760128018.

[52] P. VINCENT, H. LAROCHELLE, Y. BENGIO & P.-A. MANZAGOL, "Extracting and Composing Robust Features with Denoising Autoencoders", in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08, Helsinki, Finland: ACM, 2008, pp. 1096–1103.

[53] J. DENG, W. DONG, R. SOCHER, L. LI, KAI LI & LI FEI-FEI, "ImageNet: A large-scale hierarchical image database", in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[54] M. D. ZEILER, D. KRISHNAN, G. W. TAYLOR & R. FERGUS, "Deconvolutional networks", in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2528–2535.

[55] A. KRIZHEVSKY, I. SUTSKEVER & G. E. HINTON, "ImageNet Classification with Deep Convolutional Neural Networks", *Neural Information Processing Systems*, vol. 25, Jan. 2012.

[56] D. P KINGMA & M. WELLING, "Auto-Encoding Variational Bayes", *International Conference on Learning Representations (ICLR)*, Dec. 2013.

[57] M. D. Zeiler & R. Fergus, "Visualizing and Understanding Convolutional Networks", in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele & T. Tuytelaars, Eds., Cham: Springer International Publishing, 2014, pp. 818–833.

[58] A. Graves, G. Wayne & I. Danihelka, "Neural Turing Machines", *CoRR*, vol. abs/1410.5401, 2014. arXiv: 1410.5401.

[59] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative Adversarial Nets", in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence & K. Q. Weinberger, Eds., Curran Associates, Inc., 2014, pp. 2672–2680.

[60] M. Mirza & S. Osindero, "Conditional Generative Adversarial Nets", *CoRR*, vol. abs/1411.1784, 2014. arXiv: 1411.1784.

[61] A. Radford, L. Metz & S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks", in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[62] P. Isola, J. Zhu, T. Zhou & A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks", in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017, pp. 5967–5976.

[63] J. Zhu, T. Park, P. Isola & A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks", in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017, pp. 2242–2251.

[64] M. Arjovsky, S. Chintala & L. Bottou, "Wasserstein Generative Adversarial Networks", in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, 2017, pp. 214–223.

[65] T. Karras, S. Laine & T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks", *CoRR*, vol. abs/1812.04948, 2018. arXiv: 1812.04948.

[66] H. Zhang, T. Xu, H. Li, *et al.*, "StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1947–1962, 2019.

[67] A. Brock, J. Donahue & K. Simonyan, "Large Scale GAN Training for High Fidelity Natural Image Synthesis", in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[68] J. Chung, Ç. Gülçehre, K. Cho & Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling", *CoRR*, vol. abs/1412.3555, 2014. arXiv: 1412.3555.

[69] C. Szegedy, Wei Liu, Yangqing Jia, *et al.*, "Going deeper with convolutions", in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[70] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens & Z. Wojna, "Rethinking the Inception Architecture for Computer Vision", *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2015.

[71] C. Szegedy, S. Ioffe, V. Vanhoucke & A. A. Alemi, "Inception-v4, inception-ResNet and the Impact of Residual Connections on Learning", in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, ser. AAAI'17, San Francisco, California, USA: AAAI Press, 2017, pp. 4278–4284.

[72] K. Simonyan & A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[73] K. He, X. Zhang, S. Ren & J. Sun, "Deep Residual Learning for Image Recognition", in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[74] J. Schmidhuber, "Deep learning in neural networks: An overview", *Neural Networks*, vol. 61, pp. 85 –117, 2015.

[75] M. Jaderberg, K. Simonyan, A. Zisserman & k. kavukcuoglu, "Spatial Transformer Networks", in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama & R. Garnett, Eds., Curran Associates, Inc., 2015, pp. 2017–2025.

[76] T. D. Kulkarni, W. F. Whitney, P. Kohli & J. Tenenbaum, "Deep Convolutional Inverse Graphics Network", in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama & R. Garnett, Eds., Curran Associates, Inc., 2015, pp. 2539–2547.

[77] D. Silver, A. Huang, C. Maddison, *et al.*, "Mastering the game of Go with deep neural networks and tree search", *Nature*, vol. 529, pp. 484–489, Jan. 2016.

[78] D. Silver, J. Schrittwieser, K. Simonyan, *et al.*, "Mastering the game of Go without human knowledge", *Nature*, vol. 550, pp. 354–359, Oct. 2017.

[79] A. Graves, G. Wayne, M. Reynolds, *et al.*, "Hybrid computing using a neural network with dynamic external memory", *Nature*, vol. 538, 471 EP –, 2016, Article.

[80] S. Sabour, N. Frosst & G. E. Hinton, "Dynamic Routing Between Capsules", in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., Curran Associates, Inc., 2017, pp. 3856–3866.

[81] G. E. Hinton, S. Sabour & N. Frosst, "Matrix capsules with EM routing", in *International Conference on Learning Representations*, 2018.

[82] J. Hu, L. Shen & G. Sun, "Squeeze-and-Excitation Networks", in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[83] J. Zhou, G. Cui, S. Hu, *et al.*, "Graph neural network: A review of methods and applications", *AI Open*, vol. 1, pp. 57–81, 2020.

[84] Z. Zhang, P. Cui & W. Zhu, "Deep Learning on Graphs: A Survey", *IEEE Transactions on Knowledge & Data Engineering*, no. 01, pp. 1–1, 5555.

[85] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang & P. S. Yu, "A Comprehensive Survey on Graph Neural Networks", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, 4–24, 2021.

[86] L. Ruiz, F. Gama & A. Ribeiro, "Gated Graph Recurrent Neural Network", *IEEE Transactions on Signal Processing*, vol. 68, 6303–6318, 2020.

[87] T. N. Kipf & M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks", in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, OpenReview.net, 2017.

[88] ——, *Variational Graph Auto-Encodeurs*, 2016. arXiv: 1611.07308 [stat.ML].

[89] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò & Y. Bengio, *Graph ATtention networks*, 2018. arXiv: 1710.10903 [stat.ML].

[90] K. G. Sheela & S. N. Deepa, "Review on Methods to Fix Number of Hidden Neurons in Neural Networks", *Mathematical Problems in Engineering*, vol. 2013, Jun. 2013.

[91] I. Goodfellow, O. Vinyals & A. Saxe, "Qualitatively Characterizing Neural Network Optimization Problems", in *International Conference on Learning Representations*, 2015.

[92] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous & Y. LeCun, "The Loss Surfaces of Multilayer Networks", in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, G. Lebanon & S. V. N. Vishwanathan, Eds., ser. Proceedings of Machine Learning Research, vol. 38, San Diego, California, USA: PMLR, 2015, pp. 192–204.

[93] H. Lu & K. Kawaguchi, "Depth Creates No Bad Local Minima", *CoRR*, vol. abs/1702.08580, 2017. arXiv: 1702.08580.

[94] L. Wu, Z. Zhu & W. E, "Towards Understanding Generalization of Deep Learning: Perspective of Loss Landscapes", *CoRR*, vol. abs/1706.10239, 2017. arXiv: 1706.10239.

[95] N. Qian, "On the momentum term in gradient descent learning algorithms", *Neural Networks*, vol. 12, no. 1, pp. 145 –151, 1999.

[96] Y. E. Nesterov, "A method for solving the convex programming problem with convergence rate $O(1/k^2)$", *Dokl. Akad. Nauk SSSR*, vol. 269, pp. 543–547, 1983.

[97] J. C. Duchi, E. Hazan & Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization", *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, Jul. 2011.

[98] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method", *ArXiv*, vol. abs/1212.5701, 2012.

[99] D. P. Kingma & J. Ba, "Adam: A Method for Stochastic Optimization", in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[100] T. Dozat, "Incorporating Nesterov Momentum into Adam", in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[101] J Chavaudra & A Bridier, « [Definition of volumes in external radiotherapy: ICRU reports 50 and 62] », *Cancer radiotherapie : journal de la Societe francaise de radiotherapie oncologique*, t. 5, n° 5, 472—478, 2001.

[102] B. A. Fraass, K. P. Doppke, M. Hunt, *et al.*, "American Association of Physicists in Medicine Radiation Therapy Committee Task Group 53: quality assurance for clinical radiotherapy treatment planning.", *Medical physics*, vol. 25 10, pp. 1773–829, 1998.

[103] C. G. Orton, P. M. Mondalek, J. T. Spicka, D. S. Herron & L. I. Andres, "Lung corrections in photon beam treatment planning: are we ready?", *International Journal of Radiation Oncology*Biology*Physics*, vol. 10, no. 12, pp. 2191 –2199, 1984.

[104] K. Kappas & J. Rosenwald, "A 3-D beam subtraction method for inhomogeneity correction in high energy X-ray radiotherapy", *Radiotherapy and Oncology*, vol. 5, no. 3, pp. 223 –233, 1986.

[105] M. R. Sontag & J. R. Cunningham, "The equivalent tissue-air ratio method for making absorbed dose calculations in a heterogeneous medium", *Radiology*, vol. 129, no. 3, pp. 787–794, 1978.

[106] J. R. Cunningham, "Scatter-air ratios", *Physics in Medicine and Biology*, vol. 17, no. 1, pp. 42–51, 1972.

[107] J. R. Clarkson, "A Note on Depth Doses in Fields of Irregular Shape", *The British Journal of Radiology*, vol. 14, no. 164, pp. 265–268, 1941. eprint: https://doi.org/10.1259/0007-1285-14-164-265.

[108] M. Aspradakis, R. Morrison, N. Richmond & A. Steele, "Experimental verification of convolution/superposition photon dose calculation for radiotherapy treatment planning", *Physics in medicine and biology*, vol. 48, pp. 2873–93, Oct. 2003.

[109] A. Ahnesjö, L. Weber, A. Murman, M. Saxner, I. Thorslund & E. Traneus, "Beam modeling and verification of a photon beam multisource model", *Medical physics*, vol. 32, pp. 1722–37, Jul. 2005.

[110] A. Ahnesjo, "Collapsed cone convolution of radiant energy for photon dose calculation in heterogeneous media", *Med Phys*, vol. 16, no. 4, pp. 577–592, 1989.

[111] T. Younes, « Méthodologie pour la détermination de la dose absorbée dans le cas des petits champs avec et sans hétérogénéités pour des faisceaux de photons de haute énergie », thèse de doct., Université Toulouse 3 Paul Sabatier, 2018.

[112] R. Mohan & J. Antolak, "Monte Carlo techniques should replace analytical methods for estimating dose distributions in radiotherapy treatment planning", *Medical physics*, vol. 28, pp. 123–6, Mar. 2001.

[113] I. Chetty, B. Curran, J. Cygler, *et al.*, "Report of the AAPM Task Group No. 105: issues associated with clinical implementation of Monte Carlo-based photon and electron external beam treatment planning. Med Phys", *Medical physics*, vol. 34, pp. 4818–53, Jan. 2008.

[114] D. W. Rogers, B. A. Faddegon, G. X. Ding, C. M. Ma, J. We & T. R. Mackie, "BEAM: a Monte Carlo code to simulate radiotherapy treatment units", *Med Phys*, vol. 22, no. 5, pp. 503–524, 1995.

[115] D. Rogers, B. Walters & I. Kawrakow, "BEAMnrc users manual", *NRC Report PIRS 509(a)revH*, Jan. 2004.

[116] F. Salvat, J. Fernández-Varea & J. Sempau, "Penelope. A code system for Monte Carlo simulation of electron and photon transport", *NEA Data Bank, Workshop Proceeding, Barcelona*, pp. 4–7, Jan. 2007.

[117] J. F. Briesmeister, "MCNP: A General Monte Carlo N-Particle Transport Code", 2000.

[118] S. Agostinelli, J. Allison, K. Amako, *et al.*, "Geant4—a simulation toolkit", *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 506, no. 3, pp. 250 –303, 2003.

[119] S. Jan, D Benoit, E. Becheva, *et al.*, "GATE V6: A major enhancement of the GATE simulation platform enabling modelling of CT and radiotherapy", *Physics in Medicine and Biology*, vol. 56, p. 881, Jan. 2011.

[120] D. Sarrut, M. Bardiès, N. Boussion, *et al.*, "A review of the use and potential of the GATE Monte Carlo simulation code for radiation therapy and dosimetry applications", *Medical physics*, vol. 41, p. 064 301, Jun. 2014.

[121] J. Camilleri, « Dosimétrie in vivo des traitements de radiothérapie conformationnelle à modulation d'intensité par imageur portal haute énergie au silicium amorphe », thèse de doct., Université Toulouse 3 Paul Sabatier, 2014.

[122] R. Alecu, J. J. Feldmeier & M. Alecu, "Dose perturbations due to in vivo dosimetry with diodes", *Radiotherapy and Oncology*, vol. 42, no. 3, pp. 289 –291, 1997.

[123] A. Rizzotti, C. Compri & G. Garusi, "Dose evaluation to patients irradiated by 60Co beams, by means of direct measurement on the incident and on the exit surfaces", *Radiotherapy and Oncology*, vol. 3, no. 3, pp. 279 –283, 1985.

[124] G. Leunens, J. V. Dam, A. Dutreix & E. van der Schueren, "Quality assurance in radiotherapy by in vivo dosimetry. 2. Determination of the target absorbed dose", *Radiotherapy and Oncology*, vol. 19, no. 1, pp. 73 –87, 1990.

[125] A. Noel, P. Aletti, P. Bey & L. Malissard, "Detection of errors in individual patients in radiotherapy by systematic in vivo dosimetry", *Radiotherapy and Oncology*, vol. 34, no. 2, pp. 144 –151, 1995.

[126] S. Marcié, E. Charpiot, R.-J. Bensadoun, *et al.*, "In vivo measurements with MOSFET detectors in oropharynx and nasopharynx intensity modulated radiation therapy", *International Journal of Radiation Oncology*Biology*Physics*, vol. 61, no. 5, pp. 1603 –1606, 2005.

[127] G. P. Beyer, C. W. Scarantino, B. R. Prestidge, *et al.*, "Technical Evaluation of Radiation Dose Delivered in Prostate Cancer Patients as Measured by an Implantable MOSFET Dosimeter", *International Journal of Radiation Oncology*Biology*Physics*, vol. 69, no. 3, pp. 925 –935, 2007.

[128] L. Antonuk, J. Boudry, J. Yorkston, C. Wild, M. Longo & R. Street, "Radiation damage studies of amorphous-silicon photodiode sensors for applications in radiotherapy X-ray imaging", *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 299, no. 1, pp. 143 –146, 1990.

[129] R. A. Street, S. Nelson, L. Antonuk & V. Perez Mendez, "Amorphous Silicon Sensor Arrays for Radiation Imaging", *MRS Proceedings*, vol. 192, p. 441, 1990.

[130] L. Antonuk, "TOPICAL REVIEW: Electronic portal imaging devices: a review and historical perspective of contemporary technologies and research", *Physics in medicine and biology*, vol. 47, R31–65, Apr. 2002.

[131] J. Siebers, J. Kim, L. Ko, P. Keall & R. Mohan, "Monte Carlo computation of dosimetric amorphous silicon electronic portal images", *Medical physics*, vol. 31, pp. 2135–46, Jul. 2004.

[132] M. B., *Clinical 3D Dosimetry in Modern Radiation Therapy*. CRC Press, 2017.

[133] A. Piermattei, A. Fidanzio, L. Azario, *et al.*, "In patient dose reconstruction using a cine acquisition for dynamic arc radiation therapy", *Medical & biological engineering & computing*, vol. 47, pp. 425–33, Mar. 2009.

[134] F. Younan, « Reconstruction de la dose absorbée in-vivo en 3D pour les traitements RCMI et arcthérapie à l'aide des images EPID de transit », thèse de doct., Université Toulouse 3 Paul Sabatier, 2018.

[135] A. Mans, P. Remeijer, I. Olaciregui-Ruiz, *et al.*, "3D Dosimetric verification of volumetric-modulated arc therapy by portal dosimetry", *Radiotherapy and Oncology*, vol. 94, no. 2, pp. 181 –187, 2010, Selected papers from the 10th Biennial ESTRO Conference on Physics and Radiation Technology for Clinical Radiotherapy.

[136] L. Berger, P. Francois, G. Gaboriaud & J.-C. Rosenwald, "Performance optimization of the Varian aS500 EPID system", *Journal of applied clinical medical physics / American College of Medical Physics*, vol. 7, pp. 105–14, Feb. 2006.

[137] M. Podesta, S. Nijsten, J. Snaith, *et al.*, "Measured vs simulated portal images for low MU fields on three accelerator types: Possible consequences for 2D portal dosimetry", *Medical physics*, vol. 39, pp. 7470–9, Dec. 2012.

[138] B. McCurdy, K Luchka & S. Pistorius, "Dosimetric investigation and portal dose image prediction using an amorphous silicon electronic portal imaging device", *Medical physics*, vol. 28, pp. 911–24, Jul. 2001.

[139] P. B. Greer & C. C. Popescu, "Dosimetric properties of an amorphous silicon electronic portal imaging device for verification of dynamic intensity modulated radiation therapy", *Med Phys*, vol. 30, no. 7, pp. 1618–1627, 2003.

[140] A. V. Esch, T. Depuydt & D. P. Huyskens, "The use of an a-Si-based EPID for routine absolute dosimetric pre-treatment verification of dynamic IMRT fields", *Radiotherapy and Oncology*, vol. 71, no. 2, pp. 223 –234, 2004.

[141] C Kirkby & R Sloboda, "Consequences of the spectral response of an a-Si EPID and implications for dosimetric calibration", *Medical physics*, vol. 32, pp. 2649–58, Aug. 2005.

[142] P. Greer, "Correction of pixel sensitivity variation and off-axis response for amorphous silicon EPID dosimetry", *Medical physics*, vol. 32, pp. 3558–68, Jan. 2005.

[143] M. Grattan & C. McGarry, "Mechanical characterization of the varian Exact-arm and R-arm support systems for eight aS500 electronic portal imaging device", English, *Medical Physics*, vol. 37, no. 4, pp. 1707–13, Apr. 2010.

[144] M. Podesta, S. Nijsten, L. Persoon, S. Scheib, C. Baltes & F. Verhaegen, "Time dependent pre-treatment EPID dosimetry for standard and FFF VMAT", *Physics in Medicine and Biology*, vol. 59, pp. 4749–4768, Aug. 2014.

[145] P. McCowan, E. Uytven, T. Vanbeek, G. Asuni & B. McCurdy, "An in vivo dose verification method for SBRT–VMAT delivery using the EPID", *Medical Physics*, vol. 42, pp. 6955–6963, Dec. 2015.

[146] P. M. McCowan & B. M. McCurdy, "Frame average optimization of cine-mode EPID images used for routine clinical in vivo patient dose verification of VMAT deliveries", *Med Phys*, vol. 43, no. 1, p. 254, 2016.

[147] L. McDermott, R. Louwe, J Sonke, M Herk & B. Mijnheer, "Dose-response and ghosting effects of an amorphous silicon electronic portal imaging device", *Medical physics*, vol. 31, pp. 285–95, Feb. 2004.

[148] P. Winkler, A. Hefner & D. Georg, "Dose-response characteristics of an amorphous silicon EPID", *Medical physics*, vol. 32, pp. 3095–105, Nov. 2005.

[149] C. Yeboah & S. Pistorius, "Monte Carlo studies of the exit photon spectra and dose to a metal/phosphor portal imaging screen", *Medical physics*, vol. 27, pp. 330–9, Mar. 2000.

[150] L. Parent, J. Seco, P. Evans, A. Fielding & D. Dance, "Monte Carlo modelling of a-Si EPID response: The effect of spectral variations with field size and position", *Medical physics*, vol. 33, pp. 4527–40, Jan. 2006.

[151] S. Nijsten, W. van Elmpt, M Jacobs, *et al.*, "A global calibration model for a-Si EPIDs used for transit dosimetry", *Medical physics*, vol. 34, pp. 3872–84, Nov. 2007.

[152] A. Kavuma, M. Glegg, G. Currie & A. Elliott, "Assessment of dosimetrical performance in 11 Varian a-Si500 electronic portal imaging device", *Physics in medicine and biology*, vol. 53, pp. 6893–909, Dec. 2008.

[153] B McCurdy & P. Greer, "Dosimetric properties of an amorphous-silicon EPID used in continuous acquisition mode for application to dynamic and arc IMRT", *Medical physics*, vol. 36, pp. 3028–39, Aug. 2009.

[154] B King, L Clews & P. Greer, "Long-term two-dimensional pixel stability of EPIDs used for regular linear accelerator quality assurance", *Australasian physical & engineering sciences in medicine / supported by the Australasian College of Physical Scientists in Medicine and the Australasian Association of Physical Sciences in Medicine*, vol. 34, pp. 459–66, Dec. 2011.

[155] R. Louwe, L. McDermott, J Sonke, *et al.*, "The long-term stability of amorphous silicon flat panel imaging devices for dosimetry purposes", *Medical physics*, vol. 31, pp. 2989–95, Dec. 2004.

[156] J. M. Boudry & L. E. Antonuk, "Radiation damage of amorphous silicon photodiode sensors", *IEEE Transactions on Nuclear Science*, vol. 41, no. 4, pp. 703–707, 1994.

[157] J. M. Boudry & L. E. Antonuk, "Radiation damage of amorphous silicon, thin-film, field-effect transistors", *Med Phys*, vol. 23, no. 5, pp. 743–754, 1996.

[158] B. Mijnheer, P. González, I. Olaciregui-Ruiz, R. Rozendaal, M. Herk & A. Mans, "Overview of 3-year experience with large-scale electronic portal imaging device-based 3-dimensional transit dosimetry", *Practical radiation oncology*, vol. 5, Sep. 2015.

[159] D. Roberts, J. Moran, L Antonuk, Y. El-Mohri & B. Fraass, "Charge trapping at high doses in an active matrix flat panel dosimeter", vol. 51, Nov. 2003, 1599–1603 Vol.3.

[160] L. McDermott, S. Nijsten, J. Sonke, M. Partridge, M Herk & B. Mijnheer, "Comparison of ghosting effects for three commercial a-Si EPIDs", *Medical physics*, vol. 33, pp. 2448–51, Aug. 2006.

[161] P. Rowshanfarzad, B. McCurdy, M. Sabet, C. Lee, J. O'Connor & P. Greer, "Measurement and modeling of the effect of support arm backscatter on dosimetry with a Varian EPID", *Medical physics*, vol. 37, pp. 2269–78, May 2010.

[162] A Mans, M Wendling, L. McDermott, *et al.*, "Catching errors with in vivo EPID dosimetry", *Medical physics*, vol. 37, pp. 2638–44, Jun. 2010.

[163] G. Poludniowski, M Thomas, P Evans & S Webb, "CT reconstruction from portal images acquired during volumetric-modulated arc therapy", *Physics in medicine and biology*, vol. 55, pp. 5635–51, Oct. 2010.

[164] P. Rowshanfarzad, M. Sabet, J. O'Connor, P. McCowan, B. McCurdy & P. Greer, "Gantry angle determination during arc IMRT: Evaluation of a simple EPID-based technique and two commercial inclinometers", *Journal of applied clinical medical physics / American College of Medical Physics*, vol. 13, p. 3981, Nov. 2012.

[165] P. Rowshanfarzad, H. Riis, S. Zimmermann & M. Ebert, "A comprehensive study of the mechanical performance of gantry, EPID and the MLC assembly in Elekta LINACs during gantry rotation", *The British journal of radiology*, vol. 88, p. 20 140 581, Apr. 2015.

[166] S. Wang, J. K. Gardner, J. J. Gordon, *et al.*, "Monte Carlo-based adaptive EPID dose kernel accounting for different field size responses of imagers", *Med Phys*, vol. 36, no. 8, pp. 3582–3595, 2009.

[167] B. McCurdy & S. Pistorius, "A two-step algorithm for predicting portal dose images in arbitrary detectors", *Medical physics*, vol. 27, pp. 2109–16, Sep. 2000.

[168] T. R. McNutt, T. R. Mackie, P. Reckwerdt, N. Papanikolaou & B. R. Paliwal, "Calculation of portal dose using the convolution/superposition method", *Med Phys*, vol. 23, no. 4, pp. 527–535, 1996.

[169] R. Louwe, E. Damen, M Herk, A. Minken, O Törzsök & B. Mijnheer, "Three-dimensional dose reconstruction of breast cancer treatment using portal imaging", *Medical physics*, vol. 30, pp. 2376–89, Oct. 2003.

[170] G. Jarry & F. Verhaegen, "Patient-specific dosimetry of conventional and intensity modulated radiation therapy using a novel full Monte Carlo phase space reconstruction method from electronic portal images", *Physics in medicine and biology*, vol. 52, pp. 2277–99, Apr. 2007.

[171] L. McDermott, M. Wendling, J.-J. Sonke, M. Herk & B. Mijnheer, "Replacing Pretreatment Verification With In Vivo EPID Dosimetry for Prostate IMRT", *International journal of radiation oncology, biology, physics*, vol. 67, pp. 1568–77, Apr. 2007.

[172] W. van Elmpt, S. Nijsten, R. F. H. Schiffeleers, *et al.*, "A Monte Carlo based three-dimensional dose reconstruction method derived from portal dose images", *Medical physics*, vol. 33, pp. 2426–2434, Aug. 2006.

[173] W. van Elmpt, S. Nijsten, S. Petit, B. Mijnheer, P. Lambin & A. Dekker, "3D in vivo dosimetry using megavoltage CBCT and EPID dosimetry", *International journal of radiation oncology, biology, physics*, vol. 73, pp. 1580–7, May 2009.

[174] W. van Elmpt, S. Nijsten, A. Dekker, B. Mijnheer & P. Lambin, "Treatment verification in the presence of inhomogeneities using EPID-based three-dimensional dose reconstruction", *Medical physics*, vol. 34, pp. 2816–26, Jul. 2007.

[175] E. Uytven, T. Vanbeek, P. McCowan, K. Chytyk-Praznik, P. Greer & B. McCurdy, "Validation of a method for in vivo 3D dose reconstruction for IMRT and VMAT treatments using on-treatment EPID images and a model-based forward-calculation algorithm", *Medical Physics*, vol. 42, pp. 6945–6954, Dec. 2015.

[176] V. N. Hansen, P. M. Evans & W. Swindell, "The application of transit dosimetry to precision radiotherapy", *Med Phys*, vol. 23, no. 5, pp. 713–721, 1996.

[177] A. Ahnesjo, P. Andreo & A. Brahme, "Calculation and application of point spread functions for treatment planning with high energy photon beams", *Acta Oncol*, vol. 26, no. 1, pp. 49–56, 1987.

[178] M. Partridge, M Ebert & B. Hesse, "IMRT verification by three-dimensional dose reconstruction from portal beam measurements", *Medical physics*, vol. 29, pp. 1847–58, Sep. 2002.

[179] P. Francois, P. Boissard, L. Berger & A. Mazal, "In vivo dose verification from back projection of a transit dose measurement on the central axis of photon beams", *Physica medica : PM : an international journal devoted to the applications of physics to medicine and biology : official journal of the Italian Association of Biomedical Physics (AIFB)*, vol. 27, pp. 1–10, Jan. 2011.

[180] J. Camilleri, J. Mazurier, D. Franck, P. Dudouet, I. Latorzeff & X. Franceries, "Clinical results of an EPID-based in-vivo dosimetry method for pelvic cancers treated by intensity modulated radiation therapy", *Physica Medica*, vol. 30, Sep. 2014.

[181] A. Piermattei, S. Cilla, L. Azario, *et al.*, "a-Si EPIDs for the in-vivo dosimetry of static and dynamic beams", *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 796, Feb. 2015.

[182] S. Nijsten, B. Mijnheer, A. Dekker, P. Lambin & A. Minken, "Routine individualised patient dosimetry using electronic portal imaging devices", *Radiotherapy and oncology : journal of the European Society for Therapeutic Radiotherapy and Oncology*, vol. 83, pp. 65–75, May 2007.

[183] K. L. Pasma, M. Kroonwijk, S. Quint, A. G. Visser & B. J. Heijmen, "Transit dosimetry with an Electronic Portal Imaging Device (EPID) for 115 prostate cancer patients", *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 45, no. 5, pp. 1297–1303, 1999.

[184] A. Piermattei, A. Fidanzio, G. Stimato, *et al.*, "In vivo dosimetry by an a-Si-based EPID", *Medical physics*, vol. 33, pp. 4414–22, Dec. 2006.

[185] C.-S. Chang, Y.-H. Tseng, J.-M. Hwang, R. Shih & K.-S. Chuang, "Dosimetric characteristics and day-to-day performance of an amorphous-silicon type Electronic Portal Imaging Device", *Radiation Measurements*, vol. 91, Apr. 2016.

[186] R. Boellaard, M. van Herk & B. J. Mijnheer, "A convolution model to convert transmission dose images to exit dose distributions", *Med Phys*, vol. 24, no. 2, pp. 189–199, 1997.

[187] M. Wendling, R. Louwe, L. McDermott, J.-J. Sonke, M. Herk & B. Mijnheer, "Accurate two-dimensional IMRT verification using a back-projection EPID dosimetry method", *Medical physics*, vol. 33, pp. 259–73, Mar. 2006.

[188] M. Wendling, L. McDermott, A. Mans, J.-J. Sonke, M. Herk & B. Mijnheer, "A simple backprojection algorithm for 3D in vivo EPID dosimetry of IMRT treatments", *Medical physics*, vol. 36, pp. 3310–21, Aug. 2009.

[189] M. Wendling, L. McDermott, A. Mans, *et al.*, "In aqua vivo EPID dosimetry", *Medical physics*, vol. 39, pp. 367–77, Jan. 2012.

[190] B. King, D. Morf & P. Greer, "Development and testing of an improved dosimetry system using a backscatter shielded electronic portal imaging device", *Medical physics*, vol. 39, pp. 2839–47, May 2012.

[191] B. King & P. Greer, "A method for removing arm backscatter from EPID images", *Medical physics*, vol. 40, p. 071 703, Jul. 2013.

[192] B. Zwan, B. King, J. O'Connor & P. Greer, "Dose-to-water conversion for the backscatter-shielded EPID: A frame-based method to correct for EPID energy response to MLC transmitted radiation", *Medical physics*, vol. 41, p. 081 716, Aug. 2014.

[193] J. Camilleri, J. Mazurier, D. Franck, P. Dudouet, I. Latorzeff & X. Franceries, "2D EPID dose calibration for pretreatment quality control of conformal and IMRT fields: A simple and fast convolution approach", *Physica Medica*, vol. 32, Nov. 2015.

[194] X. Tang, T. Lin & S. Jiang, "A feasibility study of treatment verification using EPID cine images for hypofractionated lung radiotherapy", *Physics in Medicine and Biology*, vol. 54, no. 18, S1–S8, Aug. 2009.

[195] M. Nyflot, P. Thammasorn, L. S. Wootton, E. C. Ford & W. A. Chaovalitwongse, "Deep learning for patient-specific quality assurance: Identifying errors in radiotherapy delivery by radiomic analysis of gamma images with convolutional neural network", *Medical Physics*, vol. 46, no. 2, pp. 456–464, Dec. 2018.

[196] G. Kalantzis, L. A. Vasquez-Quino, T. Zalman, G. Pratx & Y. Lei, "Toward IMRT 2D dose modeling using artificial neural networks: A feasibility study", *Medical physics*, vol. 38, pp. 5807–5817, Oct. 2011.

[197] N. Childress, C. Bloch, R White, M. Salehpour & I. Rosen, "Detection of IMRT delivery errors using a quantitative 2D dosimetric verification system", *Medical physics*, vol. 32, pp. 153–62, Feb. 2005.

[198] B. Nelms & J. Simon, "A survey on planar IMRT QA analysis", *Journal of applied clinical medical physics / American College of Medical Physics*, vol. 8, p. 2448, Feb. 2007.

[199] A. Van Esch, J. Bohsung, P. Sorvari, *et al.*, "Acceptance tests and quality control (QC) procedures for the clinical implementation of Intensity Modulated Radiation Therapy (IMRT) using inverse planning and the sliding window technique: Experience from five radiotherapy departments", *Radiotherapy and oncology : journal of the European Society for Therapeutic Radiotherapy and Oncology*, vol. 65, pp. 53–70, Nov. 2002.

[200] D. Low, W. Harms, S. Mutic & J. Purdy, "A technique for the quantitative evaluation of dose distributions", English, *Medical Physics*, vol. 25, no. 5, pp. 656–661, May 1998.

[201] T. Ju, T. Simpson, J. O. Deasy & D. A. Low, "Geometric interpretation of the gamma dose distribution comparison technique: interpolation-free calculation", *Medical physics*, vol. 35, no. 3, 879—887, 2008.

[202] J. Sanders & E. Kandrot, *CUDA by Example: An Introduction to General-Purpose (GPU) Programming*, 1st. Addison-Wesley Professional, 2010.

[203] R. L. Siddon, "Fast calculation of the exact radiological path for a three-dimensional CT array", *Medical Physics*, vol. 12, no. 2, pp. 252–255, Mar. 1985.

[204] E. Sundermann, F. Jacobs, M. Christiaens, B. De Sutter & I. Lemahieu, "A Fast Algorithm to Calculate the Exact Radiological Path Through a Pixel Or Voxel Space", *Journal of Computing and Information Technology*, vol. 6, Dec. 1998.

Publications

Conférences nationales

Communications orales

[1] F. Chatrie, F. Younan, Xavier Franceries et al., « Contrôle qualité et dosimétrie *in-vivo* par imagerie portale », in *33 èmes Journées LARDs*, Besançon, France, 13-14 Octobre 2016.

[2] F. Chatrie, X. Franceries & M.-V. Le Lann, « Dosimétrie *in-vivo* et contrôle qualité en radiothérapie externe par réseaux de neurones », in *6 èmes journées des SFRPs - Codes de calcul en radioprotection, radiophysique et dosimétrie*, Sochaux, France, 1-02 Février 2018.

[3] F. Chatrie, A. R. Barbeiro, X. Franceries & M.-V. Le Lann, « Utiliser l'imageur portal avec des réseaux de neurones », in *57 èmes journées SFPM*, Toulouse, France, 13-15 Juin 2018.

Lien vidéo du congrès : http://www.canalc2.tv/video/15162

[4] F. Chatrie, X. Franceries & M.-V. Le Lann, « Contrôle qualité en radiothérapie externe basé sur une imagerie portale via des réseaux de neurones », in *5 èmes conférence APIA*, Toulouse, France, 1-05 Juillet 2019.

Poster

[5] A. R. Barbeiro, L. Parent, F. Chatrie, *et al.*, "Characterization of an a-Si-1000 EPID response in integrated and continuous acquisition modes for application to SBRT dosimetry", in *57 èmes journées SFPM*, Toulouse, France, 13-15 Juin 2018.

Conférences jeunes scientifiques

[6] F. Chatrie, X. Franceries & M.-V. Le Lann, "External radiotherapy quality control and in vivo dosimetry via artificial neural networks", in *Workshop Radiotherapy modelling*, Luz Saint-Sauveur, France, 21-24 September 2016.

[7] F. Chatrie, A. Malaroda, E. McKay & M. Bardiès, "Modelling of clinical dosimetry for targeted radionuclide therapy (TRT)", in *Workshop Bridging the gap between imaging and therapy*, Fort-Mahon, France, 11-13 September 2017.

Conférences internationales

Communications orales

Abstract

[8] F. Chatrie, F. Younan, J. Mazurier, *et al.*, "[OA236] Patient-specific quality assurance based on electronic portal imaging device via artificial neural networks", *ECMP, Physica Medica: European Journal of Medical Physics*, vol. 52, pp. 88–89, 2018.

Proceeding

[9] F. Chatrie, F. Younan, J. Mazurier, *et al.*, "Electronic Portal Imaging Device Using Artificial Neural Networks", in *IUPESM, World Congress on Medical Physics and Biomedical Engineering 2018*, L. Lhotska, L. Sukupova, I. Lacković & G. S. Ibbott, Eds., Singapore: Springer Singapore, 2019, pp. 633–636.

[10] F. Younan, J. Mazurier, F. Chatrie, *et al.*, "3D Absorbed Dose Reconstructed in the Patient from EPID Images for IMRT and VMAT Treatments", in *IUPESM, World Congress on Medical Physics and Biomedical Engineering 2018*, L. Lhotska, L. Sukupova, I. Lacković & G. S. Ibbott, Eds., Singapore: Springer Singapore, 2019, pp. 605–609.

Posters

[11] F. Chatrie, A. Vasseur, A. R. Barbeiro, *et al.*, "Varian and Elekta quality assurance using artificial neural network based on portal imaging", in *ESTRO 38*, Milan, Italy, 26-30 April 2019.

[12] A. R. Barbeiro, L. Parent, F. Chatrie, *et al.*, "Dosimetric response of a-Si-1000 EPID continuous imaging of FFF beams for in vivo 3D SBRT verification", in *ESTRO 38*, Milan, Italy, 26-30 April 2019.

[13] F. Chatrie, A. Vasseur, A. R. Barbeiro, J. Mazurier, X. Franceries & M.-V. Le Lann, "Quality assurance based on portal imaging using artificial neural network methods for Varian and Elekta LINACs", in *International Conference on the use of Computer in Radiation therapy (ICCR)*, Montreal, Canada, 17-20 June 2019.

[14] A. R. Barbeiro, L. Parent, L. Vieillevigne, *et al.*, "Automated characterisation of continuous portal imaging of radiotherapy FFF beams for SBRT verification", in *International Conference on the use of Computer in Radiation therapy (ICCR)*, Montreal, Canada, 17-20 June 2019.

[15] F. Chatrie, X. Franceries & M.-V. Le Lann, "Deep learning for regression problem applied to radiotherapy pre-treatment", in *30th International Workshop on Principles of Diagnosis DX'19*, Klagenfurt, Austria, 11-13 November 2019.

Reviewer

Rapporteur sur un article de revue internationale dans le journal BJR (British Journal of Radiology).

Titre : Dosimétrie *in-vivo* et contrôle qualité en radiothérapie externe par réseaux de neurones

Résumé

Le domaine de l'intelligence artificielle a subi un essor considérable ces dernières années, notamment avec l'arrivée du *Deep learning*. Des applications diverses peuvent bénéficier de ces modèles, à condition d'avoir suffisamment de données représentatives du système. Ces avancées scientifiques sont principalement dues aux prouesses technologiques qui amènent des capacités de calculs notables avec l'utilisation des cartes graphiques.

Les réseaux de neurones artificiels appliqués à la radiothérapie externe peuvent avoir un grand intérêt, notamment pour contrôler la qualité des traitements. Dans ce travail, une récente approche a été investiguée, basée sur des réseaux de neurones. Ceux-ci permettent de reconstruire une distribution de dose absorbée 2D à partir de l'imageur portal dont le signal est récupéré avant et pendant le traitement, respectivement, pour l'étape de contrôle qualité et de dosimétrie *in-vivo*. En appliquant des corrections sur ce signal, il est possible de vérifier que le patient a bien reçu ce qui lui avait été prescrit préalablement.

Les modèles utilisés sont, soit des réseaux de neurones multi-couches de type *feed-forward*, soit des réseaux de neurones convolutionnels. Ils ont été appliqués à différents types de traitements tels que la radiothérapie conformationnelle ou celle à modulation d'intensité. De plus, plusieurs énergies d'irradiation ainsi que différentes marques d'accélérateurs de particules ont été prises en charge. Afin d'évaluer la qualité du modèle conçu, le critère clinique γ_{index} a été utilisé. Des résultats tout à fait satisfaisants ont été obtenus pour la phase de contrôle qualité. Cependant, même si des résultats prometteurs pour la phase de dosimétrie *in-vivo* ont été montrés, il reste des améliorations à apporter pour pouvoir utiliser de tels algorithmes en routine clinique.

Abstract

The field of artificial intelligence have received a considerable amount of attention in recent years, particularly thanks to the arrival of Deep learning. A wide range of applications can benefit from these models, provided there is sufficient data that is representative of the system. These scientific advances are mainly due to technological progress that enables significant computing capabilities with the use of GPUs.

Artificial neural networks applied in the domain of external radiotherapy can be of great interest, especially for controlling the quality of treatments. In this work, a recent approach was investigated based on neural networks. These make possible the reconstruction of 2D absorbed dose distribution from the portal imager whose signal is recovered before and during the treatment, respectively, for quality assurance and *in-vivo* dosimetry. By correcting this signal, it is possible to verify that the patient has accurately received the prescribed treatment.

The models used are either feed-forward multi-layer neural networks or convolutional neural networks. They have been applied to different types of treatments, such as conformational or intensity-modulated radiotherapy. In addition, several irradiation energies as well as different particle accelerators manufacturers have been supported. In order to assess the quality of the design, the clinical criterion γ_{index} was used. Completely satisfactory results were obtained for the quality assurance phase. However, although promising result for the *in-vivo* dosimetry phase have been shown, there are still improvements to be made to be able to use such algorithms in clinical routine.