

Table des matières

Résumé

Table des matières

Table des illustrations

Introduction générale	1
Chapitre 1 : Apprentissage artificiel	
I. Introduction.....	3
1. Quelques définitions de base de l'apprentissage artificiel.....	3
1.1. Définition 1 (Webster).....	4
1.2. Définition 2 (Sim 83).....	5
1.3. Définition 3 (Mit 97).....	5
2. L'intérêt de l'apprentissage artificiel.....	5
3. Applications de l'apprentissage artificiel.....	6
4. Situation historique de l'apprentissage artificiel.....	7
5. L'apprentissage artificiel et les statistiques classiques.....	7
5.1. Les statistiques paramétriques.....	8
5.2. Les statistiques non paramétriques.....	8
II. Apprentissage automatique (Artificiel).....	9
1. Les tâches de l'apprentissage.....	9
1.1. L'apprentissage supervisé.....	9
1.2. L'apprentissage non supervisé.....	10
1.3. L'apprentissage par renforcement.....	10
1.4. Relations entre les techniques.....	11
2. L'induction.....	11
2.1. Définition.....	11
2.2. Typologie de l'induction.....	11
2.3. Problème de l'induction.....	12
2.4. Différents formalismes de l'apprentissage par induction.....	16
3. Formalisation de l'apprentissage exact.....	16
3.1. Identification de la limite.....	16
3.2. Résultats.....	17
4. Formalisation de l'apprentissage statistique.....	18
4.1. Minimisation du risque empirique.....	19
4.2. Dilemme biais variance.....	20
III. Méthodes d'apprentissage automatique.....	21
1. Méthodes d'apprentissage empirique.....	21
1.1. Apprentissage par analogie.....	21
1.2. Apprentissage par induction.....	21
2. Méthodes d'apprentissage fondé sur l'explication.....	22
IV. Approches symboliques de l'apprentissage automatique.....	23
1. Arbres de décision.....	23
1.1. L'apprentissage dans les arbres de décision.....	23
1.2. Les systèmes fondés sur les arbres de décision.....	26
2. Les algorithmes génétiques.....	29
2.1. Principes de base.....	29
2.2. Induction de règles à partir d'exemples.....	30
3. Les raisonnements fondés sur des cas "CBR".....	31
3.1. Principes de base.....	31
V. Approches connexionnistes de l'apprentissage automatique.....	31
1. Classification et propriétés.....	33

1.1. Principes de l'apprentissage connexionniste.....	33
1.2. Bases de données d'apprentissage.....	33
1.3. Méthodes d'apprentissage connexionniste.....	35
1.4. Les architectures des réseaux de neurones.....	36
1.5. Les différentes tâches d'un réseau de neurones.....	36
VI. Conclusion.....	37

Chapitre 2 : Réseaux de neurones artificiels

I. Introduction.....	38
1. Historique des réseaux de neurones.....	38
II. Neurones biologiques et formels.....	40
1. Définition.....	40
2. Neurone biologique.....	40
2.1. Description d'un neurone.....	40
2.2. La synapse.....	42
2.3. Principe de fonctionnement.....	42
3. Neurone formel.....	43
3.1. Définition.....	43
3.2. Modélisation.....	43
III. Réseaux de neurones artificiels.....	44
1. Définition.....	44
2. Objectif.....	45
3. Fonctionnement.....	45
4. Différents réseaux de neurones.....	46
4.1. Les réseaux de neurones statiques.....	46
4.2. Les réseaux de neurones dynamiques.....	47
5. Propriétés fondamentales des réseaux de neurones.....	49
5.1. L'approximation universelle.....	49
5.2. La parcimonie.....	50
6. Boîtes à outils neuronales.....	51
7. Applications des réseaux de neurones.....	52
7.1. La régression non linéaire.....	52
7.2. La modélisation.....	52
7.3. La commande de processus.....	52
7.4. La classification.....	52
IV. Apprentissage des réseaux de neurones.....	53
1. Procédure d'apprentissage d'un réseau de neurones.....	53
2. Position du problème.....	53
3. Fonction de coût.....	54
3.1. Présentation.....	54
3.2. Fonction de coût des moindres carrés.....	54
3.3. Minimisation de la fonction de coût.....	55
4. Le calcul du gradient.....	55
5. L'algorithme d'optimisation (apprentissage).....	56
V. Les différents algorithmes d'apprentissage.....	57
1. Modèles linéaires par rapport aux paramètres.....	57
2. Modèles non linéaires par rapport aux paramètres.....	58
2.1. Méthode du gradient à pas constant.....	58
2.2. Méthode du gradient à pas variable.....	59
2.3. Méthode de Newton.....	59
2.4. Méthode de quasi Newton.....	60
2.5. Méthode de Levenberg Marquardt.....	60
VI. Amélioration de l'apprentissage.....	61
1. Normalisation.....	61

2. Optimisation de l'architecture.....	62
2.1. L'approche par sélection.....	62
2.2. L'approche incrémentale.....	63
VII. Conclusion.....	64

Chapitre 3 : Généralités en cardiologie

I. Introduction.....	65
II. Le système cardiovasculaire.....	65
III. Anatomie du cœur.....	65
1. Activité électrique du cœur.....	67
2. Activité mécanique du cœur.....	68
IV. L'électrocardiogramme (ECG).....	70
1. Historique.....	70
2. Principe.....	70
3. Caractéristiques d'un battement cardiaque.....	72
4. Rôle de l'électrocardiogramme.....	75
V. Troubles du rythme cardiaque.....	75
1. Le rythme cardiaque.....	75
2. Les arythmies cardiaques.....	76
2.1. Extrasystole ventriculaire.....	76
2.2. Les arythmies à base de fréquence.....	77
2.3. Les arythmies à base de régularité.....	80
VI. Conclusion.....	86

Chapitre 4 : Résultats et interprétations

I. Introduction.....	87
II. Les bases de données médicales.....	87
1. La base de données MIT BIH.....	88
2. Base d'exemples utilisée.....	88
III. Extraction des descripteurs.....	90
1. Intervalle RRp.....	90
2. Intervalle RRs.....	90
3. Rapports des intervalles RRs/RRp.....	91
4. Largeur du complexe QRS.....	91
5. Distance PP.....	91
6. Energie du complexe QRS.....	91
IV. Apprentissage des réseaux de neurones.....	91
1. Phase d'apprentissage.....	91
2. Phase de test (validation).....	92
V. Conception des réseaux de neurones.....	93
1. Architectures.....	93
2. Fonctions d'activation.....	93
3. Algorithmes d'apprentissage.....	94
VI. Différents scénarios d'expérimentation.....	95
1. Phase d'apprentissage.....	95
1.1. Apprentissage avec la base mixte.....	95
1.2. Apprentissage avec la base des battements "N".....	99
1.3. Apprentissage avec la base des battements "V".....	101
2. Phase de test.....	103
2.1. Evaluation avec la base des battements mixte.....	103
2.2. Evaluation avec la base des battements "N".....	105
2.3. Evaluation avec la base des battements "V".....	107
VII. Conclusion.....	109

Conclusion générale.....	110
Bibliographie.....	111
Annexes	

Table des illustrations

Chapitre 1 : Apprentissage artificiel

Figure 1 : Situation de l'apprentissage automatique.....	4
Figure 2 : Exemple d'arbre de décision simple.....	24
Figure 3 : Exemple d'arbre de décision complexe.....	26
Figure 4 : Exemple de neurone biologique.....	32
Figure 5 : Exemple de neurone formel.....	32
Tableau 1 : Ensemble d'apprentissage : conditions météorologiques.....	24

Chapitre 2 : Réseaux de neurones artificiels

Figure 1 : Modèle de neurone biologique.....	41
Figure 2 : Cellule nerveuse.....	42
Figure 3 : Modèle général d'un réseau de neurone formel.....	43
Figure 4 : Réseau de neurones non bouclé.....	47
Figure 5 : Réseau de neurones bouclé.....	48
Figure 6 : L'approximateur universel.....	49

Chapitre 3 : Généralités en cardiologie

Figure 1 : Circulation artérielle et veineuse.....	66
Figure 2 : Anatomie du cœur.....	67
Figure 3 : Vue générale du cœur.....	67
Figure 4 : Les nœuds et faisceau de His.....	68
Figure 5 : Cycle mécanique cardiaque.....	69
Figure 6 : Principe de fonctionnement de l'ECG.....	72
Figure 7 : ECG normal.....	74
Figure 8 : Schémas des différents aspects du complexe QRS.....	74
Figure 9 : Le rythme cardiaque d'un étudiant pendant un examen oral.....	76
Figure 10 : Extrasystole ventriculaire.....	77
Figure 11 : Bradycardie sinusale.....	77
Figure 12 : Rythme d'échappement jonctionnel.....	78
Figure 13 : Tachycardie sinusale.....	79
Figure 14 : Tachycardie ventriculaire.....	80
Figure 15 : Emplacement des foyers ectopiques.....	81
Figure 16 : Exemple du rythme de bigéminisme.....	81
Figure 17 : Exemple du rythme de trigéminisme.....	82
Figure 18 : Fibrillation auriculaire.....	83
Figure 19 : Fibrillation ventriculaire.....	83
Figure 20 : Exemple de Mobitz de type II.....	84
Figure 21 : Exemple de bloc de branche gauche.....	85
Tableau 1 : Caractéristiques mécaniques et électriques d'une arythmie normale.....	75

Chapitre 4 : Résultats et interprétations

Figure 1 : Intervalles RRp, RRs et PP.....	90
Figure 2 : Variation de l'erreur du RN (RN10la) avec la base des battements "N".....	100
Histogramme 1 : Erreurs minimums en fonction des différents RNs.....	97
Histogramme 2 : Temps d'apprentissage minimums en fonction des différents RNs.....	99
Histogramme 3 : Erreurs d'apprentissage avec la base des battements "N".....	100
Histogramme 4 : Temps d'apprentissage avec la base des battements "N".....	101
Histogramme 5 : Erreurs d'apprentissage avec la base des battements "V".....	102
Histogramme 6 : Temps d'apprentissage avec la base des battements "V".....	102
Histogramme 7 : Taux de reconnaissance des meilleurs RNs (test avec la base mixte)...	104

Histogramme 8 : Sensibilité des meilleurs RNs.....	105
Histogramme 9 : Taux de reconnaissance des RNs (test avec la base "N").....	106
Histogramme 10 : Spécificité des RNs avec la base de test "N".....	106
Histogramme 11 : Taux de reconnaissance des meilleurs RNs (test avec la base "V")...	107
Histogramme 12 : Sensibilité des RNs avec la base "V".....	108
Tableau 1 : Quelques bases de données cardiologiques.....	87
Tableau 2 : Caractéristiques des enregistrements de la base d'exemples utilisée.....	88
Tableau 3 : Matrice des résultats de test.....	92
Tableau 4 : Exemples de fonctions d'activation.....	94
Tableau 5 : Résultats d'apprentissage avec la base de données mixte.....	95
Tableau 6 : Les RNs optimaux en valeurs d'erreurs.....	97
Tableau 7 : Les RNs optimaux en valeurs de temps d'apprentissage.....	98
Tableau 8 : Résultats obtenus en effectuant l'apprentissage avec la base "N".....	99
Tableau 9 : Résultats obtenus en effectuant l'apprentissage avec la base "V".....	101
Tableau 10 : Résultats de test avec la base mixte.....	103
Tableau 11 : Evaluation des RNs avec la base de test "N".....	105
Tableau 12 : Evaluation des RNs avec la base de test "V".....	107

Annexes

Tableau 1 : Enregistrements de la base MIT BIH.....	i
Tableau 2 : Signification des symboles utilisés.....	ii
Tableau 3 : Différents classifieurs neuronaux réalisés.....	iii
Tableau 4 : Résultats obtenus avec les classifieurs particuliers.....	iv

التلخيص:

إن طريقة التشخيص للمساعدة في أخذ القرار عملية ذات أهمية حيث يبقى التحليل البشري نموذج مثالي خاصة في الميدان الطبي. الغاية من التشخيص هي قدرة القرار بعد ملاحظة طبيعة الشيء و إلى أي صنف يشابهه، بمعنى آخر إلى أي قسم ينتمي. طريقة شبكة العصبونات طبقت في عملية التشخيص و أثبتت جدارتها منذ سنوات مضت و حققت عدة مكاسب. التعرف المسبق على مختلف الأمراض تسمح بالتكفل المناسب و تحسين نوعية حياة المرضى. أنجزنا في مذكرة الماجستير هذه دراسة واسعة و معمقة لمختلف المشاكل التقنية أثناء مرحلة التعلم الكلاسيكي في النماذج العصبونية، أخذنا بعين الاعتبار متغيرات النموذج في حد ذاته و كذلك طريقة توزيع الأمثلة في قاعدة التعلم. الكلمات المفتاحية: التعلم الاصطناعي، شبكة العصبونات، أمراض القلب، التصنيف.

Résumé :

La démarche diagnostique pour l'aide à la décision est une application privilégiée où le raisonnement humain demeure le modèle de référence notamment dans le domaine médical.

La finalité d'un diagnostic est de pouvoir décider, après avoir observé un objet, à quelle forme typique celui-ci est similaire, ou en d'autres termes, à quelle classe connue il peut être associé ou affecté. L'approche neuronale appliquée au problème du diagnostic a fait ses épreuves depuis déjà plusieurs années et présente plusieurs atouts. Une détection précoce des différentes maladies permettrait d'assurer une prise en charge adéquate et améliorer la qualité de vie des patients.

Nous réalisons dans ce mémoire de magistère une étude large et approfondie des différents problèmes d'apprentissage classique dans les modèles neuronaux en tenant compte des paramètres du modèle et de la distribution des exemples de la base d'apprentissage.

Mots clés : Apprentissage artificiel, Réseaux de neurones, Cardiologie, Classification.

Abstract

The diagnostic approach for decision support is a preferred application in which human reasoning remains the reference model especially in the medical field.

The diagnosis purpose is to decide, after observing an object, how typical form it is similar, or in other words, how can it be classed. For several years, the neural approach applied to the diagnostic problem is tried and tested and has several advantages. Early detection of diseases would ensure adequate care and improve quality of life of patients.

We realize in this master work a thorough study of various problems of classical learning in neural models respecting parameters of the model and the distribution of examples of the training set.

Keywords: Machine Learning, Neural Networks, Cardiology, Classification.

Introduction générale

Dans le cadre de ce mémoire de magistère, on traite la reconnaissance des anomalies cardiaques, par le biais d'un ensemble de classifieurs neuronaux, afin de pouvoir les améliorer par la suite.

De nos jours, les maladies cardiovasculaires comptent parmi les principales causes de mortalité dans le monde entier, et avec le vieillissement des populations, le nombre des décès d'origine cardiaque ira sans doute en croissance [World Health Statistics 2006]. Or, la majorité des décès d'origine cardiaque surviennent dans la phase pré hospitalière [Eur Heart J, 1998], [J. McMurray et al., 1994], [R. Luepker, 2005], [A. McGinn, 2005]. Une détection précoce des symptômes d'une cardiopathie permettrait d'assurer une meilleure prise en charge adéquate, et d'améliorer la qualité de vie des patients.

En effet, plusieurs études épidémiologiques ont été faites et ont montré l'impact potentiel de la durée entre l'apparition des symptômes et la prise en charge effective du patient [Eur Heart J, 1998]. En pratique, il faut trouver un moyen pour réduire cette durée afin de commencer les traitements au plus tôt. Or, le problème qui se pose n'est pas simple, même les experts (cardiologues) se trompent parfois, car l'arythmie cardiaque peut être instable (présente à domicile du patient et disparaît à l'hôpital, pour réapparaître un moment après). D'où l'intérêt des solutions d'aide au diagnostic médical, et d'aide à la décision.

Depuis un siècle environ, EINTHOVEN a introduit en médecine l'électrocardiogramme de surface (ECG) qui constitue jusqu'aujourd'hui un des piliers essentiels du diagnostic. Au cours des dernières années, les méthodes d'analyse des pathologies cardiaques se renforcent par l'introduction des procédés d'imagerie, notamment de l'échocardiogramme. Ceci ne modifie pas l'importance et l'utilité des ECGs. De l'autre côté, les connaissances apportées par traitement des troubles du rythme cardiaque.

Avec l'évolution de l'informatique et du calcul numérique, il devient intéressant d'intégrer une démarche d'aide au diagnostic dans un processus de calcul automatique dans l'élaboration des stratégies thérapeutiques à partir des signaux biomédicaux (ECG...).

L'apprentissage artificiel est devenu un domaine autonome de l'intelligence artificielle, bien situé par son côté "carrefour" : optimisation, statistiques, biologie, informatique, approximation, contrôle. L'apprentissage est un terme très général qui décrit le processus selon lequel l'être humain ou la machine peut accroître sa connaissance. Apprendre c'est donc raisonner: découvrir des analogies et des similarités, généraliser ou particulariser une expérience, tirer parti de ses échecs et erreurs passés pour des raisonnements ultérieurs. Les nouveaux acquis sont utilisés pour résoudre de nouveaux problèmes, accomplir une nouvelle tâche ou accroître les performances dans l'accomplissement d'une tâche existante.

L'apprentissage artificiel repose sur la résolution de deux problèmes : Le premier est celui du regroupement (appelé classification en analyse de données) : étant donnée une masse de connaissances, comment découvrir des ressemblances entre elles, de sorte que l'on puisse les regrouper en sous-groupes plus simples et ayant une signification. Le second problème est dit de discrimination, il représente l'apprentissage de procédures de classification : étant donné un ensemble d'exemples, comment trouver une méthode qui permet de reconnaître efficacement chaque classe de ces exemples.

Les algorithmes d'apprentissage artificiel requièrent typiquement peu de variables mais très significatifs caractérisant le phénomène étudié. Dans quelques algorithmes d'apprentissage, il est recommandé d'utiliser des modules de réduction de dimensionnalité pour éviter la redondance des données. Cela a un effet important sur la performance du système.

Les réseaux de neurones, ensembles d'opérateurs non linéaires interconnectés, forment une famille de fonctions non linéaires, qui permet de construire, par apprentissage, une très large classe de modèles. Grâce à leur facilité de mise en œuvre, les réseaux de neurones ont prouvé leur efficacité particulièrement dans le domaine de classification, précisément dans notre cas la classification des arythmies cardiaques de type battements ventriculaires prématurés [Belgacem 2002], [Hedeili 2004], [Achyara 2004].

Ce mémoire de magistère comporte quatre chapitres :

- Le premier chapitre décrit les fondements théoriques de l'apprentissage artificiel.
- Le deuxième chapitre porte sur les réseaux de neurones artificiels, ainsi que leurs propriétés fondamentales et principes de fonctionnement, et qui seront par la suite utilisés dans la classification des arythmies cardiaques
- Dans le chapitre 3, une présentation générale du fonctionnement du système cardiovasculaire, puis le principe de l'électrocardiogramme (ECG). Et à la fin une brève description des différentes anomalies et arythmies cardiaques.
- Et on présentera dans le dernier chapitre, les différents résultats de classification des battements (normaux ou battements ventriculaires prématurés) obtenus après la conception d'une variété de classifieurs neuronaux à base des réseaux de neurones multicouches, avec différentes architectures et paramètres d'apprentissage, ainsi que l'interprétation de ces résultats.

Chapitre 1 : Apprentissage artificiel

I. Introduction

En informatique, avec ses diverses disciplines, et plus précisément en intelligence artificielle ; la notion de l'apprentissage artificiel joue un rôle primordiale ; car il répond au besoin de l'intelligence artificielle (IA) qui cherchait à construire des systèmes de plus en plus autonomes, des algorithmes capables de résoudre des problèmes de forte complexité.

L'intelligence artificielle, science de la modélisation des connaissances, s'est enrichie d'une nouvelle méthodologie avec l'apprentissage artificiel, science, entre autre, de l'induction logique.

Et dans ces dernières années, l'apprentissage artificiel appelé aussi apprentissage automatique « Machine Learning », s'est imposé avec ses nombreuses méthodes et techniques dans différents domaines d'application de l'informatique.

L'apprentissage automatique est apparu grâce à la conception du développement de l'enfant, du développement de l'expertise et de l'acquisition des connaissances par instruction ou découverte.

1. Quelques définitions de base de l'apprentissage artificiel

- L'apprentissage est toute méthode permettant de construire un modèle de la réalité à partir des données, soit en améliorant partiellement le modèle, soit en créant un nouveau modèle. Il existe deux tendances principales en apprentissage, celle issue de l'intelligence artificielle et qualifiée de symbolique, et celle issue des statistiques et qualifiée de numérique.
- L'apprentissage automatique, est un champ d'étude de l'intelligence artificielle, a pour objectif d'extraire et d'exploiter automatiquement l'information présente dans un jeu de données. En cela il couvre un vaste champ d'objectifs comme la fouille de données, la classification non supervisée, la sélection de variables, la discrimination, la régression, la sélection de modèle, la génération et l'inférence de règles, etc. Il s'avère également être fortement pluridisciplinaire puisque selon les données et les objectifs, l'apprentissage automatique fait appel à l'informatique, aux neurosciences, au traitement du signal, aux sciences cognitives, à la théorie de l'information, à la biologie, et aux statistiques, au bon sens (Figure 1) .

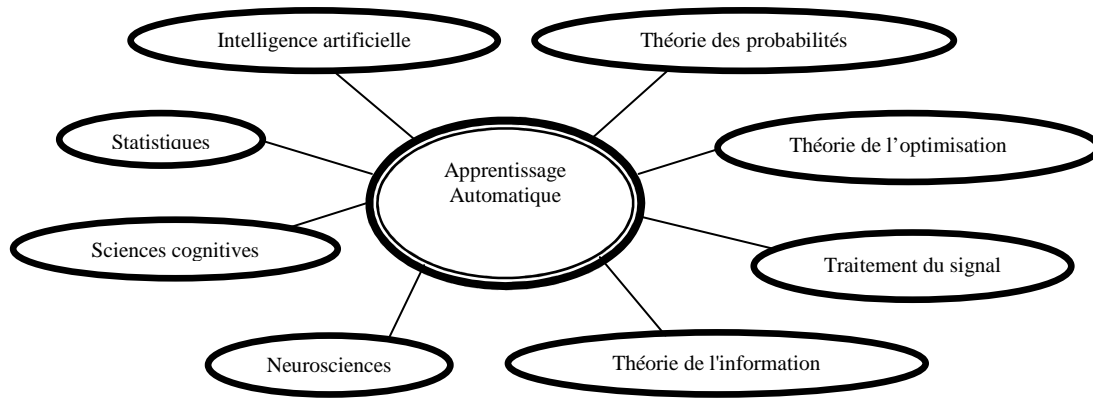


Figure 1: situation de l'apprentissage automatique.

- L'apprentissage automatique fait référence au développement, à l'analyse et à l'implémentation de méthodes qui permettent à une machine (au sens large) d'évoluer grâce à un processus d'apprentissage, et ainsi de remplir des tâches qu'il est difficile ou impossible de remplir par des moyens algorithmiques plus classiques.
- L'apprentissage automatique est un ensemble de processus permettant à un système d'accroître ses connaissances et de modifier son comportement à la suite de ses expériences et de ses actes passés.

1.1. Définition 1 (Webster)

Apprendre, c'est acquérir un savoir, une compréhension ou une aptitude par l'étude, l'instruction ou l'expérience.

Il est question d'apprentissage dans de nombreux domaines, chacun ayant une définition propre de cette notion. On peut par exemple citer les domaines suivants:

- Philosophie : le problème posé par l'apprentissage est celui de la possibilité même du raisonnement par induction dont les enjeux épistémologiques sont importants.
- Linguistique : l'apprentissage est l'acquisition de la langue maternelle en particulier apprentissage de la syntaxe en priorité sur celui de la morphologie et la sémantique.
- Psychologie : la psychologie distingue mal la mémoire de l'apprentissage et s'intéresse à l'apprentissage des catégories de base ou des concepts.
- Biologie : la biologie s'intéresse surtout au substrat matériel de l'apprentissage et assimile l'apprentissage à une tâche de contrôle (réglage d'une boucle sensori-motrice).

Si l'on se place du point de vue des sciences de l'ingénieur comme l'informatique, la robotique, l'automatique, on utilise la notion d'apprentissage automatique. Une première définition, assez générale de l'apprentissage dans ce cadre est donnée par Herbert Simon.

1.2. Définition 2 [Sim83]

L'apprentissage dénote des changements dans un système qui sont adaptatifs au sens où ils permettent à ce système d'exécuter une même tâche ou des tâches tirées de la même population plus efficacement la fois suivante.

« Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the same task or tasks drawn from the same population more efficiently and more effectively next time ».

De façon un peu plus précise, [Mit97] isole les éléments communs à tous les problèmes d'apprentissage. Trois éléments sont nécessaires pour parler d'apprentissage : tâche, mesure de performance, base d'apprentissage.

1.3. Définition 3 [Mit97]

Un programme apprend à partir d'une base d'apprentissage E par rapport à une classe de tâches T et une mesure de performance P, si sa performance à l'exécution de tâches de T, mesurée par P, s'améliore au fur et à mesure que E lui est présentée.

« A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E ».

2. Intérêt de l'apprentissage artificiel

L'apprentissage artificiel intervient si on n'a pas assez de connaissances explicites pour obtenir un algorithme qui résout le problème en question ; mais nous avons beaucoup d'exemples de la tâche à accomplir (base de données = exemples).

La plupart des programmes d'intelligence artificielle possèdent aujourd'hui un module d'apprentissage et tous les programmes de reconnaissance des formes sont fondés sur des algorithmes d'apprentissage. Actuellement, l'apprentissage artificiel permet de réaliser plusieurs tâches différentes :

- Assistance des experts humains dans la prise des décisions complexes (aide aux diagnostics médicaux, analyse des marchés financiers).
- Fouilles d'immenses bases de données hétérogènes (data mining).
- Analyse automatique des photos satellites pour détecter certaines ressources sur la Terre, ainsi que l'identification des sous-marins.
- Reconnaissance de formes et de la parole humaine et de l'interpréter.
- Recherche d'information (moteur internet...).
- Détection des fraudes dans le domaine de la télécommunication.
- Etc...

L'objectif de l'apprentissage automatique est de concevoir des machines capables d'évoluer automatiquement grâce à l'expérience [Mit97]. Pour cela, l'apprentissage automatique regroupe l'ensemble des méthodes permettant à une machine de construire un modèle de la réalité à partir des données, soit en améliorant un modèle partiel ou moins général, soit en créant complètement ce modèle [CMK02].

L'apprentissage automatique est à l'heure actuelle une discipline en plein essor et ses domaines d'application sont très nombreux.

3. Applications de l'apprentissage artificiel

Un programme est plus efficace s'il est doté d'une possibilité d'apprentissage. Reprenons pour cela les applications de l'intelligence artificielle et de la reconnaissance des formes.

- Un programme de reconnaissance de la parole augmente ses performances au fur et à mesure de son utilisation par la même personne : c'est une expérience qu'il est aujourd'hui facile de faire en pratique si nous achetons un logiciel personnel de dictée vocale.
- Un programme de détection des ressources terrestres apprend à reconnaître une zone de pollution au milieu de la mer, à partir d'une base de données d'exemples d'images de zones connues comme propres ou comme polluées : cette base de données lui sert d'expérience pour déterminer sa décision sur une zone inconnue.
- Un programme de diagnostic sur un ensemble d'informations évolutives prises sur un patient doit avoir été pourvu de connaissances, à partir de diagnostics de praticiens et d'experts sur des situations types. Mais il doit aussi avoir été doté d'un module de généralisation, de façon à réagir correctement à des situations auxquelles il n'a jamais été confronté exactement.
- Les moteurs de recherche sur le Web pourraient être munis d'un module d'adaptation au style de navigation de l'utilisateur : c'est une faculté souhaitable pour augmenter l'ergonomie de leur utilisation. Les programmes ne sont pas encore réellement agrémentés de cette propriété.
- Un programme de jeu d'échecs possède en général une très bonne efficacité a priori ; mais il est naturel d'essayer de le doter d'un module où il puisse analyser ses défaites et ses victoires, pour améliorer ses performances moyennes dans ses parties futures. Ce module d'apprentissage existe dans un certain nombre de programmes de jeux.

4. Situation historique de l'apprentissage artificiel

Traditionnellement, le domaine de l'apprentissage automatique est considéré symbolique est né vers la fin des années 50, comme une branche dissidente de l'Intelligence Artificielle classique, avec la publication des travaux de Rosenblatt sur le Perceptron [Rosenblatt 75].

Historiquement, c'est là le fruit de la rencontre de l'Intelligence Artificielle et des Neurosciences. Ce que nous avons alors appelé la branche "connexionnisme" de l'IA ambitionnait de parvenir à créer des machines capables d'intelligence en tentant d'imiter le fonctionnement des systèmes nerveux biologiques, ou tout du moins en s'inspirant fortement des connaissances sur les réseaux de neurones biologiques, et présentait un départ radical de l'approche symbolique de logique "Aristotélicienne" adoptée par l'IA classique. Ainsi ont été développés les réseaux de neurones artificiels.

Dès sa naissance, le domaine était donc nettement interdisciplinaire. Au cours des 45 années qui ont suivi, ce caractère n'a fait que s'accroître, et si l'attrait pour la stricte inspiration biologique s'est beaucoup estompé, c'est avant tout parce que des connexions profondes ont été développées avec d'autres disciplines. En effet la formalisation du domaine, son mûrissement, la compréhension théorique accrue des problèmes impliqués, se sont accompagnés d'un rapprochement avec des disciplines ayant de solides fondations mathématiques et théoriques telles que la théorie de l'information et le traitement du signal, l'optimisation non linéaire, mais surtout et de façon prépondérante ces dernières années avec le point de vue statistique.

5. L'apprentissage automatique et les statistiques classiques

Du point de vue du problème de l'apprentissage, les statistiques classiques se divisent en deux branches :

5.1. Les statistiques paramétriques

Dont le cadre suppose que l'on connaît la forme du vrai modèle qui a généré les données, ignorant seulement ses paramètres, et où il s'agit d'estimer au mieux les paramètres du dit modèle à partir d'un échantillon de données fini.

5.2. Les statistiques non paramétriques

Là, la plupart des études statistiques s'intéressent aux propriétés de convergence et consistance de l'estimateur quand le nombre d'exemples tend vers l'infini.

Les recherches en apprentissage automatique se sont quant à elles concentrées davantage sur des problèmes réels complexes, où il serait absurde de croire que l'on puisse disposer du vrai modèle, et où l'on est également loin d'avoir une quantité illimitée de données. Bien que les statistiques classiques se soient un peu intéressées à ces questions, depuis l'avènement de l'informatique ce champ d'investigation a surtout été exploré par la communauté de

l'apprentissage automatique. Par ses origines dans des domaines moins frappés de rigueur et de formalisme mathématique (la neurobiologie et l'électronique/informatique), les recherches en intelligence artificielle symbolique ont pris un chemin davantage empirique, se satisfaisant très bien de produire des modèles mathématiques comme les réseaux de neurones, du moment qu'ils fonctionnaient et donnaient de bons résultats. Dans la mesure où les modèles utilisés étaient plus complexes, les questions de sélection de modèle et du contrôle de leur capacité se sont imposées naturellement avec force. Mais on voit que, bien plus qu'une différence de fond entre les deux domaines, ce qui les sépare est une différence de culture et d'emphase : les études statistiques classiques se sont souvent autolimitées à des modèles se prêtant bien à une analyse mathématique (modèles assez simples, en faible dimension). En comparaison, la recherche en intelligence artificielle était résolument engagée sur la voie de la complexité, avec pour seule limite la capacité du matériel informatique, et poussée par le besoin de mettre au point des systèmes répondant aux problèmes concrets du moment. Néanmoins, avec le temps, le domaine de l'apprentissage automatique a mûri, s'est formalisé, théorisé, et s'est ainsi inéluctablement rapproché des statistiques, au point d'être rebaptisé apprentissage statistique.

II. Apprentissage automatique « Artificiel »

1. Les tâches de l'apprentissage

Les tâches de l'apprentissage automatique sont divisées en trois grandes familles:

- Apprentissage supervisé,
- Apprentissage non supervisé,
- Apprentissage par renforcement.

1.1. L'apprentissage supervisé

La formulation du problème de l'apprentissage supervisé est simple : on dispose d'un nombre fini d'exemples d'une tâche à réaliser, sous forme de couples (entrée, sortie désirée), et on souhaite obtenir, d'une manière automatique, un système capable de trouver de façon relativement fiable la sortie correspondant à toute nouvelle entrée qui pourrait lui être présentée.

On distingue en général trois types de problèmes auxquels l'apprentissage supervisé est appliqué. Ces tâches diffèrent essentiellement par la nature des couples (entrée, sortie) qui y sont associées :

Classification

Dans les problèmes de classification, l'entrée correspond à une instance d'une classe, et la sortie qui y est associée indique la classe. Par exemple pour un problème de reconnaissance de visage, l'entrée serait l'image bitmap d'une personne telle que fournie par une caméra, et la sortie indiquerait de quelle personne il s'agit (parmi l'ensemble de personnes que l'on souhaite voir le système reconnaître).

Régression

Dans les problèmes de régression, l'entrée n'est pas associée à une classe, mais dans le cas général, à une ou plusieurs valeurs réelles (un vecteur). Par exemple, pour une expérience de biochimie, on pourrait vouloir prédire le taux de réaction d'un organisme en fonction des taux de différentes substances qui lui sont administrées.

Séries temporelles

Dans les problèmes de séries temporelles, il s'agit typiquement de prédire les valeurs futures d'une certaine quantité connaissant ses valeurs passées ainsi que d'autres informations. Par exemple le rendement d'une action en bourse. . . Une différence importante avec les problèmes de régression ou de classification est que les données suivent typiquement une distribution non stationnaire.

1.2. L'apprentissage non supervisé

Dans l'apprentissage non supervisé il n'y a pas de notion de sortie désirée, on dispose seulement d'un nombre fini de données d'apprentissage, constituées "d'entrées", sans qu'aucun label n'y soit rattaché.

Dans ce cas, l'algorithme d'apprentissage cherche à trouver des régularités dans une collection d'exemples, puisque dans ce type d'apprentissage on ne connaît pas la classe à laquelle les exemples d'apprentissage appartiennent.

1.2.1. Estimation de densité

Dans un problème d'estimation de densité, on cherche à modéliser convenablement la distribution des données. L'estimateur obtenu $\hat{f}(x)$ doit pouvoir donner un bon estimé de la densité de probabilité à un point de test x issu de la même distribution (inconnue) que les données d'apprentissage.

1.2.2. Partitionnement

Le problème du partitionnement « clustering » est le pendant non supervisé de la classification. Un algorithme de partitionnement tente de partitionner l'espace d'entrée en un certain nombre de "classes" en se basant sur un ensemble d'apprentissage fini, ne contenant aucune information de classe explicite. Les critères utilisés pour décider si deux points devraient appartenir à la même classe ou à des classes différentes sont spécifiques à chaque algorithme, mais sont très souvent liés à une mesure de distance entre points.

1.2.3. Réduction de dimensionnalité

Le but d'un algorithme de réduction de dimensionnalité est de parvenir à "résumer" l'information présente dans les coordonnées d'un point en haute dimension ($x \in \mathbb{R}^n, n \text{ grand}$) par un nombre plus réduit de caractéristiques ($y = f(x), y \in \mathbb{R}^m, m < n$). Le but espéré est de préserver l'information "importante", de la mettre en évidence en la dissociant du bruit, et possiblement de révéler une structure sous-jacente qui ne serait pas immédiatement apparente dans les données d'origine en haute dimension. L'exemple le plus classique d'algorithme de réduction de dimensionnalité est l'Analyse en Composantes Principales (ACP).

1.3. L'apprentissage par renforcement

La particularité et la difficulté du cadre de l'apprentissage par renforcement sont que les décisions prises par l'algorithme influent sur l'environnement et les observations futures. L'exemple typique est celui d'un robot autonome qui évolue et effectue des actions dans un environnement totalement inconnu initialement. Il doit constamment apprendre de ses erreurs et succès passés, et décider de la meilleure politique à appliquer pour choisir sa prochaine action.

1.4. Relations entre les techniques

Bien entendu, les frontières entre les tâches présentées sont souples. Ainsi nous appliquons couramment, et avec succès, des algorithmes conçus pour faire de la régression à des problèmes de classification, ou bien on estime des densités dans le but de faire de la classification.

Ce que doit être noté est qu'une bonne estimation de densité permet en théorie de prendre la décision optimale concernant un problème de classification ou de régression. Mais d'un autre côté l'estimation de densité est souvent un problème plus difficile, en pratique avec un nombre fini de données d'entraînement.

2. L'induction

Dans le cadre de l'apprentissage automatique, la définition donnée de l'induction se rapproche de celle que l'on peut trouver dans les dictionnaires :

2.1. Définition « Oxford English Dictionary »

L'induction est le processus qui consiste à inférer une loi ou un principe général à partir de l'observation d'instances particulières. Plus précisément, on utilise souvent de manière équivalente les termes induction et apprentissage à base d'exemples. Comme on l'a vu, cette forme d'apprentissage peut être ou non supervisée et peut fournir comme résultat une hypothèse globale ou locale. En plus de ces distinctions, il existe un certain nombre de caractéristiques qui distinguent les différentes façons de présenter des tâches d'apprentissage à base d'exemples [Oxford English Dictionary].

2.2. Typologie de l'induction

2.2.1. Format des exemples

Les exemples peuvent, selon les cas être des entiers naturels, des mots sur un alphabet, des vecteurs à valeurs discrètes ou continues, des formules logiques...

2.2.2. Espace des hypothèses

Dans un problème d'apprentissage à base d'exemples, on se fixe un ensemble d'hypothèses candidates dans lequel on cherche la ou les hypothèse(s) compatible(s) avec les données observées. On peut chercher ces hypothèses sous la forme de fonctions booléennes, de formules logiques, de programmes, de grammaires...

2.2.3. Critères de choix des hypothèses

Une fois choisie la forme des hypothèses que l'on recherche, il faut se donner un moyen de choisir entre plusieurs hypothèses. En général, un des premiers critères de choix est l'adéquation de l'hypothèse avec les exemples observés. Cependant, il faut pouvoir faire un choix entre deux hypothèses qui seraient également en accord avec les données. On se sert pour cela d'un biais d'apprentissage qui peut avoir les formes suivantes :

- Biais de restriction : l'espace des hypothèses est restreint à un certain nombre d'hypothèses admissibles.
- Biais de préférence : un ordre de préférence sur les hypothèses est fixé à l'avance, ce qui permet d'orienter la recherche en réduisant le nombre de solutions explorées. Il faut noter que l'accord d'une hypothèse avec les données constitue un biais de préférence souvent implicite.
- Biais de validation : un moyen pour valider les hypothèses qui ont été retenues (en général en les testant sur des exemples qui n'ont pas servi à l'apprentissage) est disponible.

2.2.4. Techniques de recherche

De façon simplifiée, l'apprentissage à partir d'exemples consiste à optimiser un critère (défini par le biais) dans un espace donnée (l'espace des hypothèses). Nous pouvons donc imaginer plusieurs stratégies de recherche en fonction de la connaissance que nous avons du domaine, qui peuvent aller de la simple énumération des éléments de l'espace des hypothèses à la construction d'une hypothèse explicitement basée sur les exemples observés.

2.3. Problème de l'induction

2.3.1. Point de vue philosophique

Les avis des philosophes sur l'induction divergent, que ce soit sur sa définition ou même sur sa mise en œuvre. En effet, si la déduction est une forme de raisonnement bien maîtrisée par les scientifiques, les philosophes et les logiciens, l'induction reste une problématique.

Une des premières questions qui se pose peut être formulée ainsi [Grù94] :

Comment peut-on, à partir simplement d'un nombre n_i d'instances d'un concept généraliser et reconstruire ce concept si ce dernier porte sur un nombre infini d'instances ?

Une réponse par la négative à cette question est donnée par le philosophe David Hume (1711-1776). Pour lui, l'induction est impossible puisqu'on ne peut atteindre de conclusions qu'à partir de données et méthodes connues, donc les seules conclusions que l'on peut espérer obtenir sont logiquement contenues dans la configuration de départ. Ainsi, pour Hume, la seule forme possible d'induction est la déduction.

2.3.2. Le paradoxe de Hempel :

Supposons que l'on cherche à vérifier par l'observation si l'hypothèse « tous les corbeaux sont noirs » est valide. La méthode qui vient immédiatement à l'esprit est de chercher des corbeaux et de vérifier s'ils sont effectivement noirs, auquel cas notre hypothèse sera confirmée. « Jusqu' à ce qu'on trouve une exception qui la falsifie »

Carl Hempel suggère le paradoxe suivant : l'hypothèse considérée est logiquement équivalente à « toutes les choses qui ne sont pas noires ne sont pas des corbeaux ». Ainsi, si l'on trouve une chose qui n'est pas noire, une vache violette par exemple et que cette chose n'est pas un corbeau, alors cette seconde hypothèse sera confirmée, c'est à dire que sa plausibilité va augmenter. D'où le paradoxe :

- Paradoxe de Hempel : Une vache violette est une instance qui confirme l'hypothèse que tous les corbeaux sont noirs.

La réplique traditionnelle à ce paradoxe est de dire que de la même façon, une vache violette confirme l'hypothèse que tous les corbeaux sont blancs et que nous aboutissons alors à une contradiction puisqu'une même instance ne peut confirmer deux hypothèses opposées. Cependant, il est important de voir que le procédé n'est pas le même dans les deux cas [Blo]. Dans le premier cas, nous cherchons des choses qui ne sont pas noires, et lorsque nous en trouve une nous vérifions que ce n'est pas un corbeau. Dans le deuxième cas nous cherchons des choses qui ne sont pas blanches.

La méthode du paradoxe de Hempel, n'est finalement pas si étrange, nous l'utilisons même tous les jours.

2.3.3. Le paradoxe de Goodman

Nous voyons bien que le problème de l'induction est de faire ce passage, apparemment injustifié, d'un ensemble n_i d'instances, à une hypothèse ou théorie qui permet des prédictions sur un ensemble d'instances infini : à partir de quel nombre n d'observations de la propriété P peut-on dire que $x; P(x)$?

Le paradoxe de Goodman illustre bien cette difficulté [Mue95] : supposons que l'on tire des boules de couleur dans une urne. Ayant tiré 99 boules vertes, quelle va être notre prédiction de la couleur de la centième boule ? Il semble évident que la propriété « la boule numéro i est

verte » étant vérifiée pour 99 valeurs de i , elle a de fortes chances d'être vraie pour $i = 100$. Mais Goodman propose de considérer la propriété suivante : « la boule numéro i est verte si $i \leq 99$ et bleue sinon ».

Cette propriété aussi est vérifiée par 99 valeurs de i et il semble donc tout aussi raisonnable qu'elle est vérifiée pour $i = 100$. Il n'est donc pas fondé de prédire que la couleur sera verte ou bleue.

2.3.4. Formalisation de la généralisation

L'analogie du paradoxe de Goodman dans la théorie de l'apprentissage peut être trouvée sous la forme du théorème dit de conservation de la généralisation [Sch94], encore appelé « no free lunch theorem » [Wol92].

Ce théorème s'exprime dans le cadre où l'on présente un certain nombre d'exemples à l'apprenti, qui formule une hypothèse sur le concept à apprendre et on teste la validité de cette hypothèse sur d'autres exemples. On appelle situation d'apprentissage un triplet $S = (D; m; C)$, où D est la distribution avec laquelle on tire les exemples, m est la taille de l'échantillon d'apprentissage, et C est le concept cible. Soit un algorithme d'apprentissage L , on note $G(L; D; m; C)$ la capacité de généralisation de L dans la situation $(D; m; C)$ qui est définie comme le nombre moyen d'exemples non vus (c'est à dire qui ne figurent pas dans l'échantillon d'apprentissage) bien classés.

Ce qui signifie que si dans certaines situations, L prédit bien la classification des exemples non vus (mieux qu'aléatoirement), il aura de mauvaises performances dans un certain nombre d'autres situations. Une conséquence immédiate de cette équation est qu'il est vain d'espérer pouvoir trouver un algorithme meilleur qu'un autre dans toutes les situations : il sera meilleur dans certains cas et moins bon dans d'autres.

Si ce résultat peut sembler surprenant et décourageant, il n'en est pas moins trivial dans le sens où il est évident que si tous les concepts sont équiprobables, pour chaque concept qui classe un exemple non vu comme positif, il existe un concept qui va le classer comme négatif. Rao et al. font remarquer dans [RGS95] qu'il est préférable de considérer une version du théorème qui fait intervenir la probabilité des concepts. En effet, certains concepts (ou étiquetages des exemples) peuvent être plus probables que d'autres et un bon algorithme d'apprentissage cherchera à approcher au mieux cette distribution. L'idée assez répandue aujourd'hui est que dans les domaines courants d'application des algorithmes d'apprentissage, les concepts simples sont plus probables et qu'en conséquence, on peut espérer pouvoir construire un algorithme qui aura une bonne performance dans de tels domaines.

2.3.5. La nécessité d'un biais

Tom Mitchell [Mit80, Mit97] utilise une argumentation similaire à celle de Schaefer pour mettre en évidence le fait que si l'on ne se donne pas un moyen de choisir une hypothèse

plutôt qu'une autre, on est obligé de garder toutes les hypothèses compatibles avec l'échantillon d'apprentissage et qu'alors il ne sera pas possible de généraliser puisque pour chaque exemple non vu, il y aura autant d'hypothèses qui le classeront dans chacune des classes.

La conséquence de ce raisonnement est qu'il est nécessaire, pour être capable de généraliser un concept, de disposer d'un biais d'apprentissage, c'est à dire d'un moyen de choisir une hypothèse quand plusieurs sont en accord avec les données.

Mitchell va un peu plus loin en donnant une définition formelle du biais qui n'est pas sans rappeler la position de Hume, pour qui la seule forme possible d'induction est la déduction.

2.3.6. Le dilemme biais variance

S'il est clair que l'utilisation d'un biais conditionne la capacité de généralisation, le choix de ce biais reste une tâche délicate. Dans la pratique, on utilise beaucoup les biais de restriction qui permettent de réduire le coût de l'apprentissage (puisque l'espace de recherche est réduit). Cependant le choix ce type de biais pose un certain nombre de problèmes. Le dilemme posé par l'apprentissage à partir d'exemples vient de la difficulté qu'il y a à choisir un biais de restriction. Ce dilemme est lié de manière intuitive à un équilibre à trouver entre l'adéquation des hypothèses avec les données et la capacité de l'algorithme, que [Bur98] définit comme la possibilité qu'a un appreni d'apprendre n'importe quelle base d'apprentissage sans erreur.

- **Métaphore du dilemme biais variance**

Une machine avec une capacité trop importante est comme un botaniste ayant une mémoire photographique qui, lorsqu'on lui présente un nouvel arbre, conclut qu'il ne s'agit pas d'un arbre puisqu'il a un nombre de feuille différent de tous les arbres qu'il a pu voir auparavant; une machine avec une capacité trop faible est comme le frère paresseux du botaniste qui déclare que tout ce qui est vert est un arbre. Aucun des deux ne peut bien généraliser, au sens ici de faire des prédictions correctes.

- **Dilemme biais variance et nature du biais**

Il est important de noter que lorsqu'on parle du dilemme biais variance, on s'intéresse essentiellement à un biais de restriction. En effet, si l'on considère qu'il n'y a pas de généralisation sans biais et que l'on pose la question du choix d'un biais de restriction, il s'agit de restreindre suffisamment l'espace des hypothèses à n de pouvoir généraliser tout en évitant d'en réduire la capacité de représentation. Il faut donc garder à l'esprit que les biais de préférence n'ont rien à voir avec ce dilemme.

2.4. Différentes formalisations de l'apprentissage par induction

De même qu'il existe plusieurs façons de poser le problème de l'apprentissage par induction, il existe plusieurs façons de le formaliser, selon que les exemples soient fournis passivement ou activement (requêtes), selon que l'on modélise ou non du bruit dans les données, selon que l'on demande à l'apprenti d'apprendre exactement ou de manière approchée et selon la mesure d'évaluation que l'on choisit.

Deux formalismes importants : le premier est le paradigme ou principe de Gold [Gol67, AS83], qui est un paradigme d'apprentissage exact ; le second est l'apprentissage PAC ou apprentissage Probablement Approximativement Correct, introduit par Valiant [Val84] et qui se place dans le cadre plus général de l'apprentissage statistique.

Comme on l'a vu précédemment, l'apprentissage à partir d'exemples consiste à optimiser un critère par une recherche dans l'espace des hypothèses. Il s'agit donc de définir de quelle façon l'apprenti va converger vers un optimum. C'est le mode de convergence que les différentes formalisations définissent. Par exemple, dans le cas de l'apprentissage exact, on s'intéressera à une convergence exacte (à la limite), tandis que dans l'apprentissage PAC, on s'intéressera à une convergence en probabilité. En plus du mode de convergence, la théorie de l'apprentissage permet de définir les relations qui existent entre la vitesse de convergence, la qualité de cette convergence (proximité de l'optimum global), la complexité de l'apprentissage (en nombre d'exemples ou d'erreurs...) et la complexité de l'espace des hypothèses.

3. Formalisation de l'apprentissage exact

3.1. Identification de la limite

Initialement intéressé à la modélisation de l'apprentissage du langage par les enfants, Gold considère l'inférence inductive comme un processus infini [Gol67] dans lequel, on présente à un apprenti une suite de valeurs d'une fonction : $(i_0; f(i_0)); (i_1; f(i_1)); (i_2; f(i_2));$ et l'apprenti doit être capable de déterminer une formule de prédiction sur $f(i_{n+1})$ sans erreur.

3.1.1. Définition

On a une machine (au sens de procédure mécanique, ou algorithme ou encore un apprenti.) M qui cherche à décrire correctement une règle inconnue R . A chaque pas i , on présente à M de plus en plus d'exemples (pris dans une suite infinie E) de R et M produit une conjecture g_i sur R . S'il existe un entier m pour lequel g_m est une description correcte de R et $g_m = g_{m+1} = g_{m+2} = \dots$ alors on dit que M identifiée à la limite R sur la suite d'exemples E .

On peut considérer que M apprend à chaque pas un peu plus sur la règle inconnue R et modifie sa conjecture en conséquence. Cependant, on note que même si l'identification a eu lieu (c'est à dire qu'à partir du rang m , M fait les bonnes prédictions) M ne peut déterminer que cela s'est produit.

Ainsi, pour que l'identification ait lieu, il faut qu'il y ait convergence exacte en un temps n_i , mais le moment où cette convergence a lieu est inconnu.

3.2. Résultats

Dans [Gol67], Gold démontre un certain nombre de résultats importants.

3.2.1. Fonctions récursives primitives

On peut montrer, grâce à la technique d'identification par énumération que l'on peut identifier à la limite la classe des fonctions récursives primitives (qui est récursivement énumérable).

3.2.2. Fonctions récursives totales

On ne peut pas identifier à la limite la classe des fonctions récursives totales, à cause de l'argument suivant : Supposons que M est capable d'identifier toutes les fonctions récursives à la limite.

On peut construire une fonction récursive f en simulant M : on prend $f(0) = 0$ et supposant que l'on a construit $f(0); f(1); \dots; f(n_1)$, on simule M avec comme entrées $(0; f(0)); \dots; (n_1; f(n_1))$, on calcule la prédiction que va donner M. Cet argument par diagonalisation montre donc l'impossibilité d'identifier toute la classe des fonctions récursives totales.

3.2.3. Exemples positifs

Gold considère dans une autre façon de présenter les exemples à l'apprenti. [Gol67]. On l'appelle aujourd'hui présentation par exemple positifs (Gold utilise le terme de texte pour des exemples positifs et d'informant pour des exemples étiquetés). Elle consiste à ne présenter que des exemples $(i; f(i))$ pour lesquels $f(i) = 1$.

Un des résultats les plus importants de [Gol67] est l'impossibilité d'identifier à la limite par présentation positive une classe superfinie de langages (donc en particulier la classe des langages réguliers). Cette limitation théorique n'est pas compatible avec le fait que les enfants sont capables d'apprendre leur langue maternelle avec quasiment aucun exemple négatif à leur disposition.

On notera que l'absence d'exemples négatifs est très discutable mais l'étude de l'apprentissage par exemples positifs est malgré tout justifiée par son intérêt à la fois théorique (définition de nouveaux critères et protocoles d'apprentissage) et pratique (par exemple, le web offre un immense corpus de données positives sur le langage).

Gold envisage trois hypothèses pour expliquer la possibilité pour les enfants d'apprendre leur langue maternelle :

- La classe des langages naturels est très réduite.

- Les enfants se servent d'exemples négatifs qui ne leur sont pas explicitement donnés comme tels, ou qui peuvent être de nature sémantique (incompréhension de l'interlocuteur par exemple).
- Il existe une restriction à priori sur les textes (exemples positifs) possibles, par exemple dans leur ordre de présentation (par exemple les plus simples apparaissent en premier).

4. Formalisation de l'apprentissage statistique

Contrairement à la formulation de Gold, dans le cadre de l'apprentissage statistique, on ne demande pas à l'apprenti de trouver la bonne hypothèse mais d'en trouver une qui ne soit pas trop mauvaise dans le sens où elle ne fera pas trop d'erreurs en moyenne. La différence essentielle venant du fait que le critère de succès est probabilisé. En effet, les instances sont présentées à l'apprenti après un tirage aléatoire selon une loi de probabilité fixée et, en fonction du tirage particulier réalisé, on peut réussir ou échouer (selon des critères prédéfinis) avec une certaine probabilité.

De manière plus précise, l'apprentissage statistique se présente souvent sous la forme d'un modèle général de l'estimation fonctionnelle [Vap95] :

- Un générateur G , fourni des instances (*exemples*). x éléments d'un ensemble X , appelé ensemble des instances ou domaine, et tirés indépendamment selon une distribution inconnue $D(x)$.
- Un superviseur S renvoie une valeur y , élément du codomaine Y pour toute entrée x selon une distribution $D(y, x)$ fixe et inconnue.
- Un apprenti A qui implémente un ensemble noté H de fonctions de X dans Y .

L'espace des hypothèses H est donc ici une classe de fonctions ($H \subseteq Y^X$). Dans certains cas, on suppose que cette classe est paramétrée, c'est à dire $H = \{f_x \mid f(x)\}$ [Vap95]. Les termes « hypothèse » et « fonction » sont utilisés indifféremment pour désigner les éléments de H . De même on utilisera indifféremment, dans le cadre de l'apprentissage statistique, les termes « classe de fonctions », « espace d'hypothèses » ou « ensemble d'hypothèses » pour qualifier H .

Lorsque les hypothèses prennent leurs valeurs dans un ensemble fini, on appelle ces valeurs des classes. Dans le cas où ces valeurs sont booléennes, on appelle ces hypothèses des concepts.

Usuellement, on considère la distribution produit $D(x;y) = D(y|x)D(x)$ et l'on se place dans l'espace des échantillons de longueur l muni de la mesure produit que l'on désigne par D^l (ou parfois simplement D).

Il est possible mais pas nécessaire que le superviseur S produise y à l'aide d'une des fonctions de l'espace des hypothèses ($y = f(x)$). Dans ce cas, l'apprenti aura à retrouver quelle est cette fonction (éventuellement par l'intermédiaire du paramètre correspondant à cette fonction).

Dans les autres cas, il lui faudra trouver une fonction qui donne une bonne approximation du phénomène représenté par S .

4.1. Minimisation du risque empirique

Les fonctions de l'espace H sont donc utilisées pour approcher un phénomène [Bou92] observable à partir de tirages aléatoires selon une distribution D inconnue. Afin d'estimer l'erreur commise par une certaine fonction f , on introduit une fonction de perte (Loss function) $L(y; f(x))$ qui mesure l'écart entre la prédiction de l'apprenti et la valeur attendue (donnée par le superviseur). L'erreur (ou risque) associée à une hypothèse peut alors être définie.

- Le but de l'apprentissage est alors de trouver la fonction qui minimise le risque.

On note que minimiser ce critère revient à effectuer une recherche dans l'espace des hypothèses et qu'ainsi, le choix d'une fonction de perte particulière n'est autre qu'un biais de préférence alors que le choix d'un espace de fonctions particulier représente le biais de restriction.

Vapnik [Vap95] montre par exemple que l'on peut exprimer diverses méthodes d'apprentissage (ou, dans ce cas d'estimation de dépendance) dans ce formalisme, en choisissant diverses fonctions de perte (maximum de vraisemblance, maximum à posteriori, description de longueur minimale...)

Toute la difficulté de l'apprentissage dans ce contexte vient du fait que la distribution selon laquelle sont tirés les exemples est inconnue et que le critère à minimiser concerne tout l'espace $X \times Y$ auquel l'apprenti n'a accès que par un moyen fini. En effet la seule information à laquelle a accès l'apprenti est un échantillon d'un seul élément de $X \times Y$ tiré selon D .

Tout le problème est de montrer qu'en minimisant une estimation du risque on va pouvoir minimiser le risque, au moins en probabilité. Cette estimation est donnée par la méthode de minimisation du risque empirique (ERM)[Vap95].

On peut alors se poser plusieurs questions dans ce cadre [Vap95] :

- Sous quelles conditions le principe ERM donne-t-il un apprentissage consistant ? c'est-à-dire qui donne une faible valeur de risque (réel).
- Quelle est la vitesse de convergence du processus d'apprentissage?
- Comment peut-on contrôler la vitesse de convergence (la capacité de généralisation) de l'apprentissage ?
- Comment construire des algorithmes qui peuvent contrôler la capacité de généralisation?

4.2. Dilemme biais variance

Dans le cadre de l'apprentissage statistique, le dilemme biais variance se traduit, en considérant de façon intuitive que le biais est représenté par le risque minimum atteignable avec l'espace de fonctions H .

En effet, avec un espace d'hypothèses donné on ne pourra faire mieux (à cause de la restriction de cet espace). Pour ce qui est de la variance, elle correspond à la différence entre le risque empirique et le risque réel. En effet, c'est cette différence qui rend la généralisation incertaine : avec une forte variance, on peut tomber sur un échantillon qui ne fait pas ressortir les caractéristiques du phénomène que l'on essaie d'approcher et on risque alors de choisir une hypothèse qui est très différente de la bonne.

III. Méthodes d'apprentissage automatique

Les différentes méthodes d'apprentissage automatique sont classées en deux groupes : les méthodes d'apprentissage empirique (Empirical Learning) et les méthodes d'apprentissage fondées sur l'explication (Explanation Based Learning) [TOW 91].

1. Méthodes d'apprentissage Empirique

Les méthodes d'apprentissage empirique sont fondées sur l'acquisition de connaissances à partir d'exemples. Elles incluent aussi des méthodes connues sous les noms d'Instance based learning ou Exemplar based learning. On peut citer comme exemples de méthodes d'apprentissage empirique les techniques suivantes : le raisonnement fondé sur des cas, les méthodes de construction de concepts et prototypes, les réseaux de neurones artificiels, les arbres de décision, les algorithmes génétiques d'induction de règles ou les méthodes du type ILP. Ces méthodes se divisent entre les méthodes d'apprentissage par analogie et les méthodes d'apprentissage par induction.

1.1. Apprentissage par analogie

Les approches fondées sur l'analogie essayent de faire le transfert des connaissances sur une tâche bien connue vers une autre moins connue. Ainsi, il est possible d'apprendre des nouveaux concepts ou de dériver des nouvelles solutions à partir de concepts et solutions similaires connues. Ainsi, deux notions deviennent très importantes dans la définition de l'apprentissage par analogie : le transfert et la similarité. Ce type d'approche est aussi dénommée apprentissage fondé sur la similarité.

Les approches empiriques fondées sur l'analogie, telles que les systèmes de raisonnement fondé sur des cas (CBR), peuvent être considérées plutôt comme des méthodes de raisonnement que comme des méthodes d'apprentissage. Certains de ces systèmes implémentent uniquement des fonctions de calcul de similarité, sans utiliser de mécanismes d'adaptation.

1.2. Apprentissage par Induction

L'apprentissage par induction reste toujours une des principales méthodes étudiées dans le domaine de l'apprentissage automatique. Dans cette approche, on cherche à acquérir des règles générales qui représentent les connaissances obtenues à partir d'exemples. Les règles ainsi obtenues peuvent être représentées d'une façon explicite (facilement interprétables) ou d'une façon implicite avec un codage qui n'est pas toujours facile à interpréter. L'algorithme d'apprentissage par induction reçoit un ensemble d'exemples d'apprentissage et doit produire des règles de classification, permettant de classer les nouveaux exemples. Le processus d'apprentissage cherche à créer une représentation plus générale des exemples, selon une méthode de généralisation de connaissances. Ce type de méthodes est aussi appelé apprentissage de concepts ou bien acquisition de concepts. Parmi les approches d'apprentissage empirique par induction les plus connues, on trouve les réseaux de neurones artificiels et les arbres de décision.

L'algorithme d'apprentissage par induction peut fonctionner de façon supervisée ou non supervisée.

2. Méthodes d'apprentissage fondées sur l'Explication

Les méthodes inductives, telles que les réseaux de neurones ou les arbres de décision, ont besoin d'un nombre significatif d'exemples pour pouvoir bien généraliser les connaissances (induire des règles ou des concepts). Ceci restreint les possibilités d'application de ces méthodes, puisqu'on n'a pas toujours une base d'exemples assez grande et complète sur le domaine traité.

Les méthodes d'apprentissage fondées sur l'explication (Explanation-Based Methods - EBL) [MIT 93, NIL 97] utilisent des connaissances préexistantes et un raisonnement déductif pour augmenter l'information fourni par des ensembles d'exemples. Ces méthodes sont connues sous le nom d'apprentissage par analyse (Analytical Learning) [MIT 97].

Dans les méthodes EBL, les connaissances sont dérivées à partir d'un simple cas par explication des raisons pour lesquelles il représente un exemple du concept appris. La méthode EBL utilise les connaissances préexistantes pour analyser, ou expliquer, comment chaque exemple observé lors de l'apprentissage satisfait les concepts existants. Ensuite, cette explication est utilisée pour différencier les attributs pertinents de l'exemple d'apprentissage de ceux qui ne le sont pas. De cette façon, l'exemple pourra être généralisé par un raisonnement logique, à la place des raisonnements statistiques souvent utilisés par les autres méthodes. Ces méthodes servent donc à améliorer les performances du système, grâce à des traitements qui rendent l'utilisation des connaissances plus efficace. Les méthodes d'apprentissage fondées sur l'explication considèrent que les connaissances préexistantes sont à la fois correctes et complètes.

IV. Approches symboliques de l'apprentissage automatique

Dans ce qui suit, les méthodes d'apprentissage automatique symbolique qui seront présentées, sont très connues dans le domaine de l'I.A. Le but est de mieux connaître les algorithmes, leurs avantages ainsi que leurs problèmes particuliers, afin de pouvoir proposer par la suite des solutions plus performantes.

Tout d'abord, notre concentration est basée sur trois méthodes principales : les arbres de décision, les algorithmes génétiques d'induction de règles et les systèmes de type CBR.

1. Arbres de décision

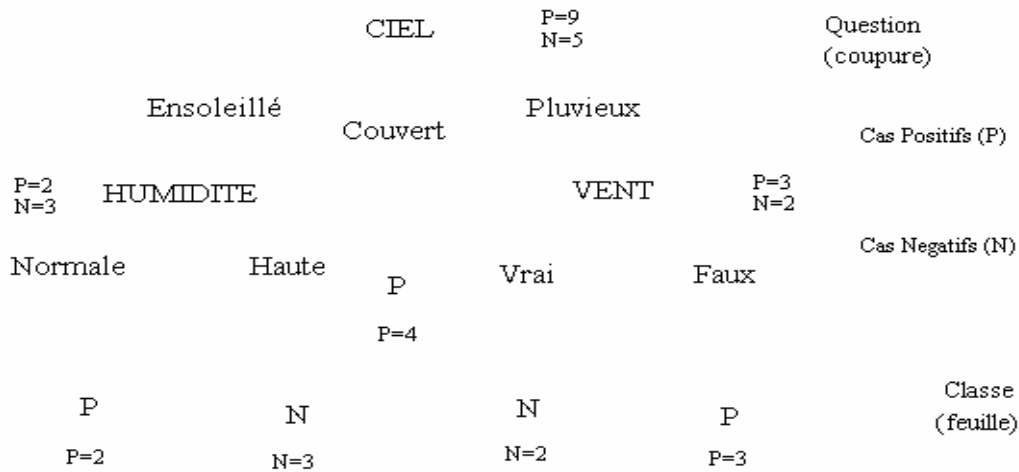
Les arbres de décision sont composés d'une structure hiérarchique en forme d'arbre. Cette structure est construite grâce à des méthodes d'apprentissage par induction à partir d'exemples.

L'arbre ainsi obtenu représente une fonction qui fait la classification d'exemples, en s'appuyant sur les connaissances induites à partir d'une base d'apprentissage. En raison de cela, ils sont aussi appelés arbres d'induction (Induction Decision Trees). Une définition un peu plus formelle des arbres de décision est la suivante : un arbre de décision est un graphe orienté, sans cycles, dont les noeuds portent une question, les arcs des réponses, et les feuilles des conclusions, ou des classes terminales.

1.1. L'Apprentissage dans les Arbres de Décision

Traditionnellement, un arbre de décision se construit à partir d'un ensemble d'apprentissage par raffinements de sa structure. Un ensemble de questions sur les attributs est construit afin de partitionner l'ensemble d'apprentissage en sous-ensembles qui deviennent de plus en plus petits jusqu'à ne contenir à la fin que des observations relatives à une seule classe. Les résultats des tests forment les branches de l'arbre et chaque sous-ensemble en forme les feuilles. Le classement d'un nouvel exemple se fait en parcourant un chemin qui part de la racine pour aboutir à une feuille :

L'exemple appartient à la classe qui correspond aux exemples de la feuille. La Figure 2 donne un exemple d'arbre de décision pour le classement d'un ensemble de cas, avec un test d'appartenance à une classe. Dans ce cas particulier, les cas dits positifs sont ceux qui appartiennent à la classe et les cas dits négatifs sont ceux qui n'y appartiennent pas.



```

IF ( ( CIEL=Ensoleillé Et HUMIDITE=Normale ) Ou
    ( CIEL=Couvert ) Ou
    ( CIEL=Pluvieux et VENT=Faux ) )
ALORS Classe = P

```

Figure 2 : Exemple d'arbre de décision simple

L'apprentissage est fait à partir d'une base d'exemples qui possèdent un certain nombre d'attributs significatifs, ex. la température, le vent, l'humidité, etc. Chaque exemple associe des valeurs particulières à chaque attribut, et comme cette méthode est une méthode d'apprentissage supervisé, chaque exemple est associé à une classe particulière. Nous présentons ci-dessous une base d'exemples afin de mieux expliquer le processus de construction d'un arbre (ce tableau est extrait de l'article de Quinlan [QUI 86]).

NUMERO	CIEL	TEMPERATURE	HUMIDITE	VENT	CLASSE
1	ensoleillé	élevé	forte	non	N
2	ensoleillé	élevé	forte	oui	N
3	couvert	élevé	forte	non	P
4	pluvieux	moyenne	forte	non	P
5	pluvieux	basse	normale	non	P
6	pluvieux	basse	normale	oui	N
7	couvert	basse	normale	oui	P
8	ensoleillé	moyenne	forte	non	N
9	ensoleillé	basse	normale	non	P
10	pluvieux	moyenne	normale	non	P
11	ensoleillé	moyenne	normale	oui	P
12	couvert	moyenne	forte	oui	P
13	couvert	élevé	normale	non	P
14	pluvieux	moyenne	forte	oui	N

Tableau 1: Ensemble d'Apprentissage : Conditions Météorologiques

Le principe de construction des arbres est le suivant : on choisit un attribut parmi les attributs non sélectionnés, et on crée un nœud portant un test sur cet attribut. Pour chaque classe d'équivalence ainsi induite, on opère le traitement suivant : si tous les exemples de cette classe

d'équivalence appartiennent à la même classe (les classes météorologiques décrites dans le tableau ci-dessus), alors on crée une feuille correspondante à cette classe, reliée au test précédent par un arc étiqueté par la valeur de l'attribut correspondant ; si tous les exemples de la classe d'équivalence considérée ne sont pas dans la même classe, alors on réitère ce processus en enlevant l'attribut précédemment considéré des attributs à sélectionner.

Par exemple, on peut construire de la manière suivante l'arbre de la Figure 2: le premier attribut sélectionné est le 'ciel'; cet attribut sépare la base en trois ensembles : "ciel ensoleillé", "ciel pluvieux" et "ciel couvert". Ensuite, pour les exemples qui ont l'attribut "ciel ensoleillé", on prend un autre attribut qui va permettre de les distinguer, par exemple l'humidité. Une fois que tous les exemples qui ont l'attribut "humidité forte" appartiennent à la même classe et qu'il en est de même avec ceux qui ont l'attribut "humidité normale", on peut arrêter le processus de division des branches de l'arbre à ce niveau. Il faut alors terminer la construction de l'arbre pour les autres branches qui n'ont pas encore été traités jusqu'à la fin.

On peut remarquer que l'ordre dans lequel ces attributs sont sélectionnés est primordial pour la construction de l'arbre. On peut avoir différents arbres résultant de l'apprentissage d'une même base d'exemple, comme le montre la Figure 7. En effet, si l'on commence par prendre la 'température', alors l'arbre considéré sera plus grand. On suppose que les arbres qui généralisent le mieux sont les arbres de petite taille, puisque ils vont tout d'abord sélectionner les décisions plus importantes et qui séparent le plus grand nombre de cas différents. De plus, la préférence pour la construction d'arbres plus simples est une application du principe du "rasoir d'Occam" [BLU 87].

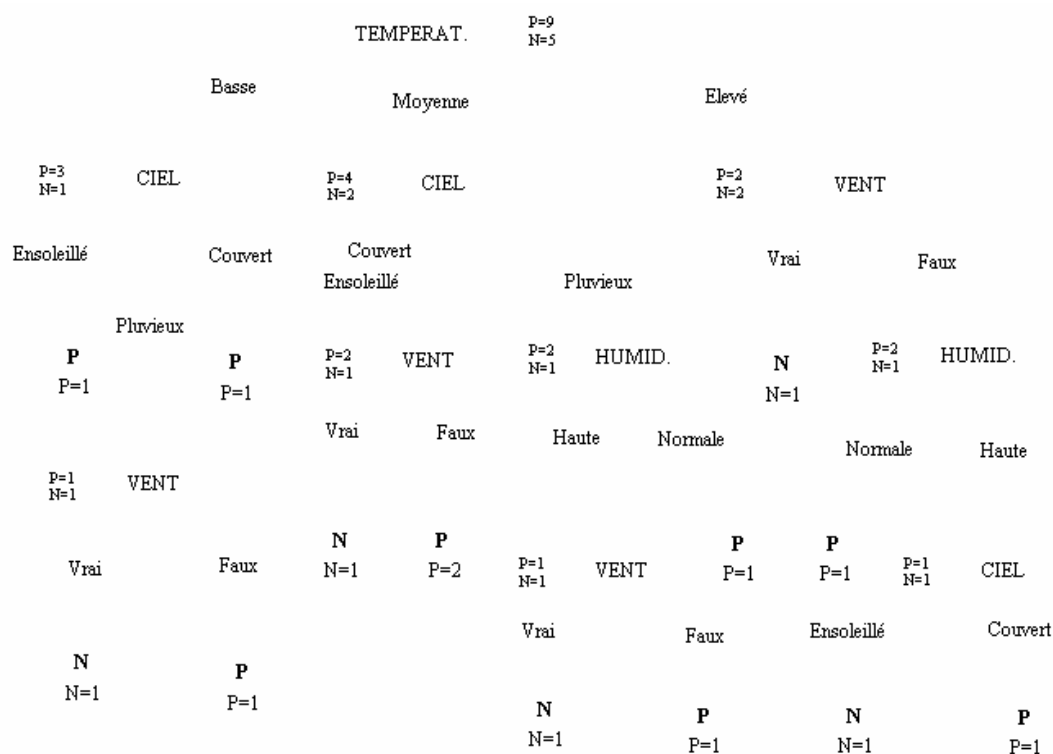


Figure 3: Exemple d'arbre de décision complexe

Quinlan a proposé une méthode pour traiter ce problème du choix de l'attribut. Il utilise une technique fondée sur la théorie de l'information de Shannon [QUI 86]. L'idée consiste à appliquer une méthode qui permet de maximiser le gain d'information apporté par chaque test.

1.2. Les Systèmes fondés sur les Arbres de Décision

Parmi les méthodes d'apprentissage fondées sur les arbres de décision les plus connues, on trouve l'algorithme ID3 [QUI 79, QUI 86] et la méthode CART [BRE 84]. Ces systèmes sont assez semblables ; ils ont été développés par deux groupes de recherche séparés et presque à la même époque. La principale différence entre ces deux systèmes réside dans le choix de la mesure utilisée pour la sélection des attributs pendant la construction de l'arbre. Cette mesure est généralement fondée sur la théorie de l'information de Shannon (entropie et gain d'information, utilisées dans ID3).

La méthode ID3 est à l'origine de plusieurs autres systèmes. L'un des plus connus est le système C4.5, développé plus récemment par Quinlan [QUI 93]. A l'origine du système C4.5 se trouve la volonté de résoudre les différents problèmes rencontrés dans son prédécesseur, l'algorithme ID3. Ces problèmes sont assez caractéristiques des arbres de décision en général, à savoir :

- a.** Les premiers systèmes ont été conçus pour traiter des attributs avec des valeurs nominales et discrètes (variables qualitatives) et ne pouvaient pas traiter des attributs avec des valeurs continues (variables quantitatives). Il manquait des méthodes bien adaptées pour trouver les bonnes questions à poser sur les valeurs continues et qui seront postérieurement associées aux nœuds de l'arbre.
- b.** Le choix du meilleur attribut qui va diviser les exemples présente lui aussi certains problèmes. Comment peut-on nous assurer que la méthode employée va nous amener à la construction de l'arbre le plus simple possible ? Malheureusement, on ne peut pas être sûr de l'obtenir, puisque ce type de problème devient intraitable d'un point de vue calculatoire [SHA 90]. De plus, il n'est pas difficile de trouver des exemples pour lesquels une variable a un très grand pouvoir de discrimination, mais n'aide pas beaucoup à la solution du problème (Ex.: l'âge ou le sexe d'une personne permet de discriminer les gens d'une façon très efficace ; toutefois est-elle une bonne variable par rapport à tous les types de problèmes de classification ?).
- c.** Parfois on ne connaît pas toutes les valeurs de chacun des attributs considérés pour la classification. Le fait que l'on doive travailler sur des domaines représentés par des informations incomplètes ouvre la problématique du traitement des attributs avec valeurs manquantes.
- d.** Les arbres de décision qui résultent de l'algorithme décrit ci-dessus souffrent du problème de sur apprentissage (overtraining/overfitting). Le fait que la construction de

l'arbre de décision ne s'arrête que lorsque tous les exemples de la feuille appartiennent à une même classe, suppose des ensembles d'apprentissage contenant uniquement des exemples "parfaits" (ils doivent être corrects et doivent bien représenter tout l'espace des entrées). L'arbre de décision finit par trop se spécialiser dans les exemples d'apprentissage, ce qui peut poser de graves problèmes au niveau de la généralisation aux nouveaux cas. Souvent, les bases d'apprentissage sont incomplètes et ont des exemples incorrects, et donc, cette méthode d'apprentissage ne fonctionnera pas correctement.

- e. Sur un arbre de décision, les connaissances restent "cryptées". Bien qu'il s'agisse d'une représentation symbolique des connaissances, elle peut être parfois un peu difficile à lire et à interpréter.
- f. Les arbres de décision, tels que ID3, doivent être complètement reconstruits pour prendre en compte un nouvel exemple ajouté à la base d'apprentissage. Ce type d'arbre n'est pas incrémental du point de vue de l'acquisition de données, même si sa structure est construite d'une façon incrémentale (par ajout de nœuds). Il faut recommencer tout l'apprentissage pour prendre en compte un nouvel exemple.
- g. Malgré le fait de que ces méthodes cherchent à construire des arbres simples, certaines branches peuvent être répliquées. La structure en arbre n'est pas toujours la forme la plus simple et économique pour représenter les informations. Les graphes de décision restent une alternative à la structuration en forme d'arbres.
- h. Les arbres de décision, comme leur nom l'indique, servent à obtenir une classification des données et ont donc une ou plusieurs sorties binaires. On ne peut pas avoir une sortie qui représente une valeur continue. Ce type d'arbres de décision n'est pas utilisable pour l'approximation de fonctions (régression).
- i. Enfin, ces arbres de décision n'exploitent pas de méthodes permettant l'utilisation des connaissances théoriques disponibles sur le problème. Dans leur forme initiale, les arbres d'induction travaillent uniquement sur les connaissances empiriques.

Plusieurs systèmes fondés sur les arbres de décision ont pris en compte ces différentes remarques décrites ci-dessus, en essayant d'apporter des solutions plus performantes :

- **C4.5** : c'est un système dérivé de l'ID3. Il présente des propositions pour traiter et améliorer les items A (discrétisation des variables quantitatives), B (prise en compte des coûts associés au choix de chaque attribut), C (prise en compte par la fonction de sélection d'attributs), D (élagage de l'arbre à la fin du processus d'apprentissage, à partir d'un ensemble d'exemples de test de généralisation) et E (explicitation de règles symboliques du type Si-Alors à partir des arbres de décision) [QUI 93, QUI 96].

- **Assistant Professional** : il utilise une autre fonction de choix d'attributs qui permet l'élagage de l'arbre de décision (post-pruning), et implémente une méthode de binarisation des attributs avec des valeurs continues [CES 87, THR 91];
- **ID5R** : Il s'agit d'une méthode de construction incrémentale d'arbres de décision qui permet d'apprendre des nouveaux exemples sans avoir besoin de recommencer tout l'apprentissage [UTG 89].
- **ITI** : système d'apprentissage incrémental d'arbres de décision, développé à partir de l>ID5R. Il apporte des améliorations par rapport à l'utilisation de variables continues et de valeurs manquantes, et il permet aussi d'élaguer les arbres (virtual pruning process) [UTG 95].
- **IDL** : Il s'agit d'un algorithme de construction incrémental d'arbres de décision fondé sur la méthode ID5R [VAN 90, THR 91].
- **PRISM** : méthode d'apprentissage fondée sur l'algorithme ID3 de Quinlan. Il se distingue d>ID3 par le type de méthode employé pour la sélection des attributs discriminants [CEN 87, THR 91].
- **SIPINA** : système de construction de graphes de décision. Il utilise une fonction particulière de sélection d'attributs, implémente une méthode de discrétisation de variables continues, implémente une méthode qui évite le sur apprentissage et permet aussi l'explicitation de règles représentées dans les graphes de décision [ZIG 92, ZIG 96b, ZIG 96c].

2. Les Algorithmes Génétiques

La notion d'Algorithme Génétique (AG) a été introduite par John H. Holland en 1975 [HOL 75] et a été considérablement développée au cours des années quatre-vingt [DAV 91, MIC 92, GOL 94]. Plus récemment, une autre voie de recherche, la programmation génétique [KOZ 92], a été dérivée des AG. On va se focaliser sur les algorithmes génétiques et leur application à l'apprentissage de règles par induction à partir d'une base d'exemples.

2.1. Principes de Base

Les algorithmes génétiques sont des algorithmes d'exploration fondés sur les mécanismes de la sélection naturelle et de la génétique. Ils utilisent à la fois les principes de survie des structures les mieux adaptées, et de modification pseudo aléatoire d'informations, pour former un algorithme d'exploration qui possède des caractéristiques intéressantes. Les AG ont une capacité de s'adapter et de se transformer sans avoir à imposer beaucoup de restrictions sur la recherche d'une solution, ce qui leur permet de mieux explorer le problème et de trouver des solutions originales.

Le principe de l'algorithme est simple et repose sur un codage des problèmes et de leurs solutions sous la forme de chaînes d'éléments de base. Les chaînes peuvent être rompues entre

chaque élément de base, à l'image des chromosomes, qui eux constituent de véritables listes de caractéristiques d'un individu. Le codage prend habituellement la forme d'une chaîne binaire très structurée, de longueur fixe ou variable selon le type de problème. On génère tout d'abord une population de solutions potentielles à un problème donné, sous la forme de telles chaînes, puis on sélectionne, au moyen d'une mesure d'ajustement, les éléments de la population qui satisfont au mieux les contraintes (ou caractéristiques) de la solution recherchée. Des opérateurs génétiques sont ensuite appliqués à cette population de manière à obtenir une nouvelle population, possédant, dans son ensemble, de meilleures solutions que la précédente génération. On réitère le processus sur plusieurs générations, jusqu'à l'obtention d'une génération de solutions qui satisfait les critères de qualité imposés, et qui est beaucoup plus adapté au traitement du problème en question. Il suffit alors de choisir parmi la population la meilleure solution.

Les AG mettent en œuvre des mécanismes assez simples pour l'évolution vers la solution. Ils ne font que des copies et des échanges de morceaux de chaînes, en plus des opérations d'évaluation et de sélection des éléments d'une population. Les manipulations des chaînes sont faites en utilisant trois opérateurs basiques : la reproduction, le croisement et la mutation.

2.2. Induction de Règles à partir d'Exemples

L'utilisation des algorithmes génétiques dans les systèmes d'apprentissage automatique s'est largement développé à partir des années quatre-vingt (80). Plusieurs applications des AG à la construction de classifieurs ont été déjà expérimentées avec des bons résultats. La première application a été réalisée par celui qui est à l'origine même des AG, J. Holland [HOL 86, HOL 87]. Plusieurs autres expérimentations ont fait suite à ces premiers travaux [DAV 91, GOL 89, DEJ 88, DEJ 90, DEJ 93b, GRE 93, EIC 96].

Pour pouvoir réaliser un apprentissage de règles avec des AG, il est nécessaire d'avoir une représentation interne de l'espace de recherche. Le codage utilisé par les AG est donc très important vis-à-vis de l'implémentation des outils d'apprentissage automatique. Le choix d'un type de codage spécifique doit essayer de garder les propriétés des AG, sans imposer des restrictions qui peuvent nuire ou restreindre la convergence de l'algorithme vers une bonne solution.

Deux approches assez connues pour générer automatiquement un classifieur à partir d'un AG sont : l'approche de Pitt et l'approche de Michigan [DEJ 93a]. L'approche de Pitt, développée initialement à l'Université de Pittsburgh, postule qu'un élément (chromosome) correspond à une base de règles et qu'une population de chromosomes est un ensemble de bases de règles. L'autre approche, celle de Michigan, a été développée à l'Université de Michigan par l'équipe de J.Holland. Cette approche postule qu'un chromosome représente une règle et qu'une

population correspond donc à une base de règles. L'algorithme, au cours de ses itérations cherche à construire et améliorer une seule base de connaissances.

3. Le Raisonnement Fondé sur des Cas - CBR

Le raisonnement fondé sur des cas (CBR) recouvre un ensemble de méthodes de résolution de problèmes qui exploite les expériences passées, plutôt que les connaissances générales d'un niveau supérieur, telles que les règles de production. Les CBR sont des méthodes d'apprentissage totalement fondées sur les connaissances empiriques, au lieu de faire usage des connaissances théoriques d'un domaine.

3.1. Principes de base

Un système CBR est capable d'utiliser la connaissance spécifique contenue dans son expérience passée pour résoudre les nouveaux problèmes. Cette expérience est représentée normalement sous la forme de cas. Ces cas, qui ont été corrigés et assignés par l'expert aux classes auxquelles ils appartiennent, constituent ainsi la mémoire d'un système CBR.

Quand un nouveau problème est présenté à un système CBR, il va se rappeler des cas passés stockés dans son mémoire, similaires au problème courant. Ensuite, le système adapte la meilleure solution mémorisée et la transfère au problème actuel. Le cas nouveau, qui a été traité par le système et reconnu, peut être à son tour mémorisé et donc ajouté comme une nouvelle expérience du système [MAL 96, KOL 93].

En général, un système de raisonnement fondé sur des cas contient les phases suivantes :

- a. Remémoration des cas les plus similaires par rapport au cas posé en question.
- b. Réutilisation de la connaissance du (ou des) cas remémoré(s) pour la résolution du problème.
- c. Révision de la solution donnée afin de la valider.
- d. Mémorisation de cette nouvelle expérience, pour une utilisation future.

V. Approches connexionnistes de l'apprentissage automatique

Les réseaux connexionnistes sont des assemblages fortement connectés d'unités de calcul, les neurones formels. Ces derniers ont pour origine un modèle du neurone biologique, dont ils ne retiennent d'ailleurs qu'une vision fort simplifiée (voir Figure 4 et Figure 5). Le neurone, comme toute cellule, est composé d'un corps (ou soma), qui contient son noyau et où se déroulent les activités propres à sa vie cellulaire. Cependant, il est aussi doté d'un axone et de dendrites, structures spécialisées dans la communication avec les autres neurones. Cette communication entre cellules nerveuses s'effectue via des impulsions nerveuses. Les impulsions sont générées à l'extrémité somatique de l'axone et vont vers les terminaisons axonales. Là, elles affecteront tous les neurones reliés au neurone générateur, par

l'intermédiaire de jonctions entre les terminaisons axonales et les autres cellules. Cette jonction est appelée synapse.



Figure 4 : Exemple de Neurone Biologique

Cet héritage de la neurobiologie forme une composante importante de l'étude des réseaux connexionnistes, et le souci de maintenir une certaine correspondance avec le système nerveux humain a animé et continue à animer une part importante des recherches dans ce domaine. Malgré cet héritage, l'essentiel des travaux d'aujourd'hui ont pour objet les réseaux de neurones formels et non son corrélat neurobiologique. Vu comme des systèmes de calcul, les réseaux de neurones possèdent plusieurs propriétés qui les rendent intéressants d'un point de vue théorique, et fort utiles en pratique. C'est cette approche - plus technique que biologique - qui est adoptée pour les travaux de recherche actuels.

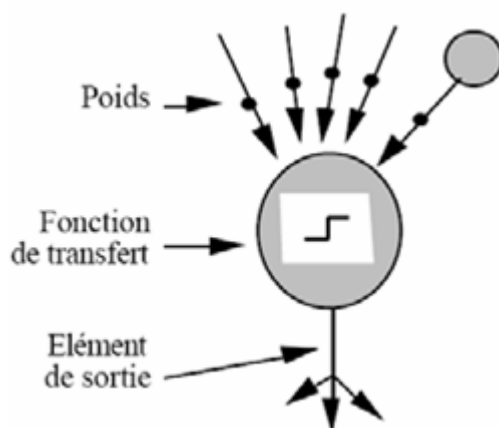


Figure 5 : Exemple de Neurone Formel

Un réseau connexionniste est constitué par un graphe orienté et pondéré. Les nœuds de ce graphe sont des automates simples nommés neurones formels ou tout simplement unités du réseau. Les unités sont dotées d'un état interne, que l'on appelle état d'activation. Les unités peuvent propager son état d'activation aux autres unités du graphe en passant par des arcs pondérés appelés connexions, liens synaptiques ou bien poids synaptiques. La règle qui

détermine l'activation d'un neurone en fonction de l'influence venue de ses entrées et de leurs poids respectifs s'appelle règle d'activation ou fonction d'activation. Les changements apportés aux valeurs des poids synaptiques ou dans la structure d'interconnexion des unités du réseau sont responsables des changements de comportement d'activation de ce réseau, ce qui lui permet d'apprendre un nouveau comportement. Ainsi, le réseau est capable d'établir des associations entrée-sortie (stimulus et réponse) afin de bien résoudre un problème. La méthode utilisée pour modifier le comportement d'un réseau s'appelle règle d'apprentissage.

1. Classification et propriétés

La grande quantité de modèles connexionnistes existants [SAR 97, JOD 94b] nous oblige tout d'abord à les classer selon leurs différentes propriétés, afin de mieux comprendre les avantages et les inconvénients du choix d'un modèle plutôt que l'autre. Il n'existe pas une seule façon de classer les modèles existants, mais plusieurs, selon les attributs choisis, tels que : le type d'apprentissage, l'architecture d'interconnexion des unités du réseau, la forme pour traiter et représenter les données, etc.

1.1. Principes de l'apprentissage connexionniste

L'apprentissage est en général un processus graduel, itératif, où les poids du réseau sont modifiés plusieurs fois selon une règle d'apprentissage avant d'atteindre leurs valeurs finales. Les approches d'apprentissage connexionniste peuvent être réparties en trois grandes classes, selon les types d'apprentissage déjà cités (supervisé et non supervisé).

1.2. Bases de données d'apprentissage

L'apprentissage connexionniste nécessite en général une grande quantité de données, que l'on regroupe dans un ensemble d'exemples d'apprentissage. Selon la technique d'apprentissage utilisée, d'autres ensembles de données sont aussi employés, notamment pour mesurer la validité de la solution trouvée par le réseau. On appelle ces données supplémentaires des ensembles d'exemples de test ou de généralisation. On appelle généralisation la capacité du réseau à réagir correctement lorsqu'on lui présente des entrées qui n'ont pas été vues lors de l'apprentissage. Dans la pratique, ce sont les capacités de généralisation d'un réseau connexionniste qui vont établir les possibilités de son application à différents problèmes.

Un réseau peut trop se spécialiser à un ensemble d'apprentissage. Ce type de comportement d'apprentissage va probablement mener à un problème de mauvaise généralisation, ou sur apprentissage (overfitting). Une façon d'éviter le sur apprentissage est l'utilisation de la procédure de la validation croisée (cross-validation). Cette procédure consiste en diviser les données en deux ensembles distincts : un ensemble d'apprentissage et un ensemble de test.

L'ensemble de test nous permet de contrôler l'erreur de généralisation ; ainsi, on peut déterminer le bon moment pour arrêter le processus d'apprentissage. La Figure 10 montre deux courbes où sont représentées les erreurs de sortie d'un réseau par rapport aux ensembles d'apprentissage et de test.

L'apprentissage d'un ensemble de données peut être réalisé de différentes façons, selon la manière dont le réseau est alimenté par les données :

1.2.1. Apprentissage Instantané

L'ensemble de données d'apprentissage est analysé une seule fois et permet d'établir les poids du réseau d'un seul coup. Ce mode d'apprentissage requiert un seul passage sur les données et il est aussi appelé one-shot-learning. Ce n'est pas une méthode très utilisée.

1.2.2. Apprentissage par Paquets

L'ensemble des données d'apprentissage est présenté au réseau plusieurs fois, de façon à optimiser les poids du réseau et minimiser l'erreur en sortie. Chaque présentation de l'ensemble complet de données d'apprentissage est appelée une époque (epoch). L'algorithme d'apprentissage réduit petit à petit l'erreur de sortie à chaque présentation des données, et l'état du réseau doit converger vers la solution du problème. On peut aussi manipuler l'ordre des exemples de l'ensemble d'apprentissage, ce qui peut avoir des conséquences sur l'évolution de l'apprentissage (cf. l'apprentissage actif). Cette méthode est connue aussi comme batch-learning et constitue l'un des types d'apprentissage les plus utilisés

1.2.3. Apprentissage Continu

L'algorithme d'apprentissage prend en compte continuellement les exemples qui lui arrivent (continuous/on-line learning). Cette méthode est aussi appelée apprentissage incrémental (par rapport aux données), mais nous allons utiliser plutôt le terme apprentissage continu pour éviter toute confusion avec l'apprentissage incrémental dans le cas où des unités sont ajoutées à la structure du réseau. Un des principaux problèmes de l'apprentissage continu est la difficulté de trouver un bon compromis entre adaptation et stabilité, ce qui peut amener le réseau à 'oublier' les données apprises (trop d'adaptation) ou bien être incapable de s'adapter aux nouvelles données qui arrivent (trop de stabilité).

1.2.4. Apprentissage Actif

Cette approche est basée sur l'idée que l'algorithme d'apprentissage peut lui aussi imposer des choix par rapport aux données à utiliser. L'apprentissage passe d'un comportement passif, où le réseau ne peut pas intervenir sur les données qui lui arrivent, à un comportement actif, où le réseau peut déterminer quelles données vont être prises en compte et aussi dans quel ordre elles doivent être considérées. Il s'agit d'une voie de recherche relativement récente.

1.3. Méthodes d'apprentissage connexionnistes

L'apprentissage peut être aussi implémenté par différentes méthodes, selon le type de règle d'apprentissage sélectionné. Les règles d'apprentissage les plus utilisées sont [JOD 94b, SIM 90, CAU 92] :

- **Les méthodes de correction d'erreur**, telles que la descente de gradient sur une surface d'erreur (Adaline, Perceptron, Rétro-propagation, Cascade- Correlation).
- **Les méthodes d'apprentissage par renforcement.**
- **Les méthodes d'apprentissage par compétition ou par auto-organisation**, (cartes de Kohonen).
- **Les méthodes d'apprentissage par création de prototypes ou de noyaux**, (RBF, ART).
- **Les méthodes d'apprentissage basées sur des mémoires associatives** « auto-associatives ou hétéro-associatives », (Modèle de Hopfield).
- **Les méthodes d'apprentissage temporel** (réseaux récurrents).

Il existe des méthodes qui peuvent appartenir à plus d'une classe au même temps. Par exemple, les réseaux avec apprentissage du type ARN2 incluent dans cette approche des techniques d'apprentissage non-supervisé, supervisé, par compétition et aussi par création de prototypes. Dans ce mémoire de Magistère, nous allons nous focaliser plutôt sur les méthodes d'apprentissage supervisé fondées sur la descente du gradient. Le système que nous avons développé dans le cadre de ce travail utilise un algorithme dérivé de la Rétro-Propagation [FAH 90]. Par la suite, les algorithmes d'apprentissage de type Rétro-Propagation seront étudiés avec plus de détail.

1.4. Les architectures des réseaux de neurones

La façon d'interconnecter les unités d'un réseau va nous permettre d'obtenir différentes architectures. Les architectures les plus importantes sont :

1.4.1. Réseaux à une seule couche

Les unités sont toutes sur le même niveau. Dans ce type d'architecture, les unités sont connectées directement aux entrées et sont aussi les sorties du réseau. Les réseaux à une seule couche ont normalement des connexions latérales (entre les neurones d'une même couche). Un exemple de ce type d'architecture est les réseaux du type "Kohonen Feature Map".

1.4.2. Réseaux à couches unidirectionnels

Les unités sont organisées en plusieurs niveaux bien définis, que l'on appelle des couches. Chaque unité d'une couche reçoit ses entrées à partir de la couche précédente et envoie ses sorties vers la couche suivante (feed-forward nets).

1.4.3. Réseaux récurrents

Les réseaux récurrents [SZI 95, JOD 94b, LAB 93] peuvent avoir une ou plusieurs couches, mais leur particularité est la présence d'interconnexions depuis la sortie d'une unité vers une autre unité de la même couche ou d'une couche inférieure. Ce type d'interconnexions

permet de modéliser des aspects temporeux et des comportements dynamiques, où la sortie d'une unité dépend de son état antérieur. Les boucles internes rendent ce type de réseaux instable, ce qui nous oblige à utiliser des algorithmes plus spécifiques (et usuellement plus complexes) pour l'apprentissage. Un type particulier de réseau récurrent est les réseaux totalement connectés, tels que le modèle de Hopfield.

1.4.4. Réseaux d'ordre supérieur

Les unités de ce type permettent la connexion directe entre deux entrées ou plus, avant d'appliquer la fonction de calcul de l'activation de l'unité [FIE 94a]. Ce type de réseau sert à modéliser les synapses de modulation, c'est-à-dire quand une entrée peut moduler (agir sur) le signal provenant d'une autre entrée. Un modèle particulier de réseaux d'ordre supérieur (high-order neural net) est le réseau Sigma-Pi proposé par Rumelhart [RUM 86a].

1.5. Les différentes tâches d'un réseau de neurones

Les réseaux peuvent être appliqués à plusieurs types de tâches, telles que : la reconnaissance de formes (ex. reconnaissance de visages), la classification, la transformation de données (ex. compression), la prédiction (ex. prédiction de séries temporelles), le contrôle de processus et l'approximation de fonctions. Toutes ces tâches peuvent être regroupées en deux groupes principaux selon le type des sorties fournies par le réseau et le comportement qui est recherché. Ces deux groupes principaux d'applications sont :

1.5.1. Réseaux pour l'Approximation de Fonctions

Ce type de réseau doit avoir une sortie continue et sera employé pour l'approximation exacte (interpolation) ou pour l'approximation approchée d'une fonction représentée par les données d'apprentissage [CHE 97]. Ce type de réseau est capable d'apprendre une fonction de transformation (ou d'association) des valeurs d'entrée vers les valeurs de sortie. Cette fonction acquise par le réseau permet de prédire les sorties étant données les entrées. On appelle ce type de problème, un problème de régression [BIS 97]. En général, les fonctions représentées sont des fonctions avec des variables d'entrée et de sortie continues.

1.5.2. Réseaux pour la Classification

Ce type de réseau doit attribuer des classes (valeur de sortie non-continue) aux exemples qui lui sont fournis. La classification est un cas particulier de l'approximation de fonctions où la valeur de sortie est discrète et appartient à un ensemble limité de classes. Cet ensemble de classes peut être connu d'avance dans le cas de l'apprentissage supervisé. Un réseau adapté à la classification doit avoir des sorties discrètes ou implémenter des méthodes de discrétisation des sorties (ex. application d'un seuil de discrimination).

Il est un peu délicat de classer précisément tous les différents modèles de réseaux de neurones dans une seule de ces deux classes. La plupart des modèles peuvent être adaptés pour être

utilisés dans l'un ou l'autre type d'application. Mais certains modèles sont plus adaptés à une tâche de classification que d'approximation et vice versa.

VI. Conclusion

Dans ce chapitre, nous avons essayé de faire un balayage sur tout ce qui concerne l'apprentissage artificiel, en définissant ses différentes applications, tâches et typologies, ainsi que sa situation dans les sciences cognitives et l'intelligence artificielle.

L'apprentissage artificiel se repose sur deux grandes familles d'approches : les approches symboliques (les arbres de décision, les algorithmes génétiques,...) et les approches connexionnistes (réseaux de neurones) ; ces approches ont été décrites avec leur principes de base.

Les études sur l'apprentissage artificiel sont en plein essor car c'est un domaine d'actualité qui répond à différents besoins dans la vie quotidienne.

Dans le chapitre suivant, nous effectuons une étude détaillée des réseaux de neurones.

Chapitre 2 : Réseaux de Neurones Artificiels

I. Introduction

Les réseaux de neurones sont composés d'éléments simples (ou neurones) fonctionnant en parallèle. Ces éléments ont été fortement inspirés par le système nerveux biologique. Comme dans la nature, le fonctionnement du réseau de neurones est fortement influencé par la connexion des éléments entre eux. On peut entraîner un réseau de neurones pour une tâche spécifique (reconnaissance de caractères par exemple) en ajustant les poids (ou valeurs des connexions) entre les éléments (neurones).

En général, l'apprentissage des réseaux de neurones est effectué de sorte que pour une entrée particulière présentée au réseau corresponde une cible spécifique. L'ajustement des poids se fait par comparaison entre la réponse du réseau (ou sortie) et la cible, jusqu'à ce que la sortie corresponde (au mieux ?) à la cible. On utilise pour ce type d'apprentissage dit supervisé un nombre conséquent de pair entrée/sortie.

L'apprentissage permet aux réseaux de neurones de réaliser des tâches complexes dans différents types d'application (classification, identification, reconnaissance de caractères, de la voix, vision, système de contrôle...). Ces réseaux de neurones peuvent souvent apporter une solution simple à des problèmes encore trop complexes ne pouvant être résolus rapidement par les ordinateurs actuels (puissance de calcul insuffisante) ou par notre manque de connaissances.

1. Historique des réseaux de neurones

W. James, célèbre psychologue américain introduit le concept de mémoire associative en 1890, et propose ce qui deviendra une loi de fonctionnement pour l'apprentissage sur les réseaux de neurones connue plus tard sous le nom de loi de Hebb.

Le champ des réseaux neuronaux va démarrer en 1943 par la découverte par W. McCulloch et W. Pitts du neurone formel qui est une abstraction du neurone physiologique. Le retentissement va être énorme. Par cette présentation, ils veulent démontrer que le cerveau est équivalent à une machine de Turing, la pensée devient alors purement des mécanismes matériels et logiques.

Une machine de Turing se résume à une tête de lecture comportant un nombre fini d'états internes et à un ruban. La puissance de l'analyse de Turing (1912-1954) tient au fait que sa tête de lecture ne lit qu'un seul symbole à la fois, et que cette lecture, associée à la table d'états adéquate, suffit à effectuer toutes les opérations possibles. La Machine de Turing est toutefois une machine imaginaire, abstraite, et idéale. Elle n'a pas pu être construite. Son fonctionnement implique en effet d'avoir un ruban extensible à volonté donc infini! La

combinaison de cette mémoire infinie et d'un nombre d'états fini a cependant apporté la preuve que, si un problème est calculable, alors il existe une machine pour le résoudre. Dès lors, une MACHINE, reprenant les règles de toutes les autres, est UNIVERSELLE.

En 1949, D. Hebb physiologiste américain présente dans son ouvrage «The Organization of Behavior» une règle d'apprentissage. De nombreux modèles de réseaux aujourd'hui s'inspirent encore de la règle de Hebb.

En 1957, F. Rosenblatt développe le modèle du perceptron. C'est un réseau de neurones inspiré du système visuel, et l'applique au domaine de la reconnaissance de formes. Il possède deux couches de neurones : une couche de perception et une couche liée à la prise de décision. C'est le premier système artificiel capable d'apprendre par expérience.

En 1960, le modèle de L'Adaline (ADaptive LINar Element) a été présenté par B. Widrow et Hoff. Dans sa structure, le modèle ressemble au Perceptron, cependant la loi d'apprentissage est différente. Ce modèle sera par la suite le modèle de base des réseaux multicouches.

En 1969, M. Minsky et S. Papert publient une critique des propriétés du Perceptron. Limitations alors connues, notamment concernant l'impossibilité de traiter par ce modèle des problèmes non linéaires. Ils étendent implicitement ces limitations à tous modèles de réseaux de neurones artificiels. Leur objectif est atteint, il y a abandon financier des recherches dans le domaine (surtout aux U.S.A.), les chercheurs se tournent principalement vers l'IA et les systèmes à bases de règles.

Jusqu'en 1972, où T. Kohonen présente ses travaux sur les mémoires associatives et propose des applications à la reconnaissance de formes.

C'est en 1982 que J. Hopfield présente son étude d'un réseau complètement rebouclé, dont il analyse la dynamique.

En 1983, la Machine de Boltzmann est le premier modèle connu apte à traiter de manière satisfaisante les limitations recensées dans le cas du perceptron. Mais l'utilisation pratique s'avère difficile, la convergence de l'algorithme étant extrêmement longue (les temps de calcul sont considérables).

1985 : La rétro propagation de gradient apparaît. C'est un algorithme d'apprentissage adapté aux réseaux de neurones multicouches (aussi appelés Perceptrons multicouches). Sa découverte réalisée par trois groupes de chercheurs indépendants indique que "la chose était dans l'air". Dès cette découverte, la réalisation d'une fonction non linéaire d'entrée/sortie sur un réseau en décomposant cette fonction en une suite d'étapes linéairement séparables est devenue possible. De nos jours, les réseaux multicouches et la rétro propagation de gradient restent le modèle le plus étudié et le plus productif au niveau des applications.

II. Neurones biologiques et formels

1. Définition

Un réseau de neurones est une fonction paramétrée qui est la composition d'opérateurs mathématiques simples appelés neurones formels pour les distinguer des neurones biologiques.

2. Neurone biologique

2.1. Description d'un Neurone

Les neurones sont des unités cellulaires de relais et d'interprétation de signaux situés dans le système nerveux et cérébrale de l'humain en un nombre approximatif de 25 milliards à la naissance. Ils sont composés d'un corps cellulaire appelé le soma, où se situent le noyau et toute la machinerie cellulaire, et d'extrémités telles que les dendrites et l'axone [Schmidt, 1985]. C'est le noyau qui est le site de la réaction d'atteinte ou non du seuil lorsque le neurone est stimulé et l'axone est la portion en extrémité qui fait le lien avec d'autres neurones, d'où la notion de relais, par le biais des dendrites des autres neurones. Fonctionnellement, lorsque des facteurs externes (des hormones, des neurotransmetteurs, une odeur, le touché, etc.) viennent en contact avec les récepteurs situés sur la surface des dendrites du neurone il y a modification physico-chimique dans le soma de la cellule et atteinte ou non du seuil d'inactivité de la cellule neuronale ce qui va provoquer l'activité de celle-ci si le seuil est atteint et va activer d'autres neurones par la libération de neurotransmetteurs à l'extrémité de son axone.

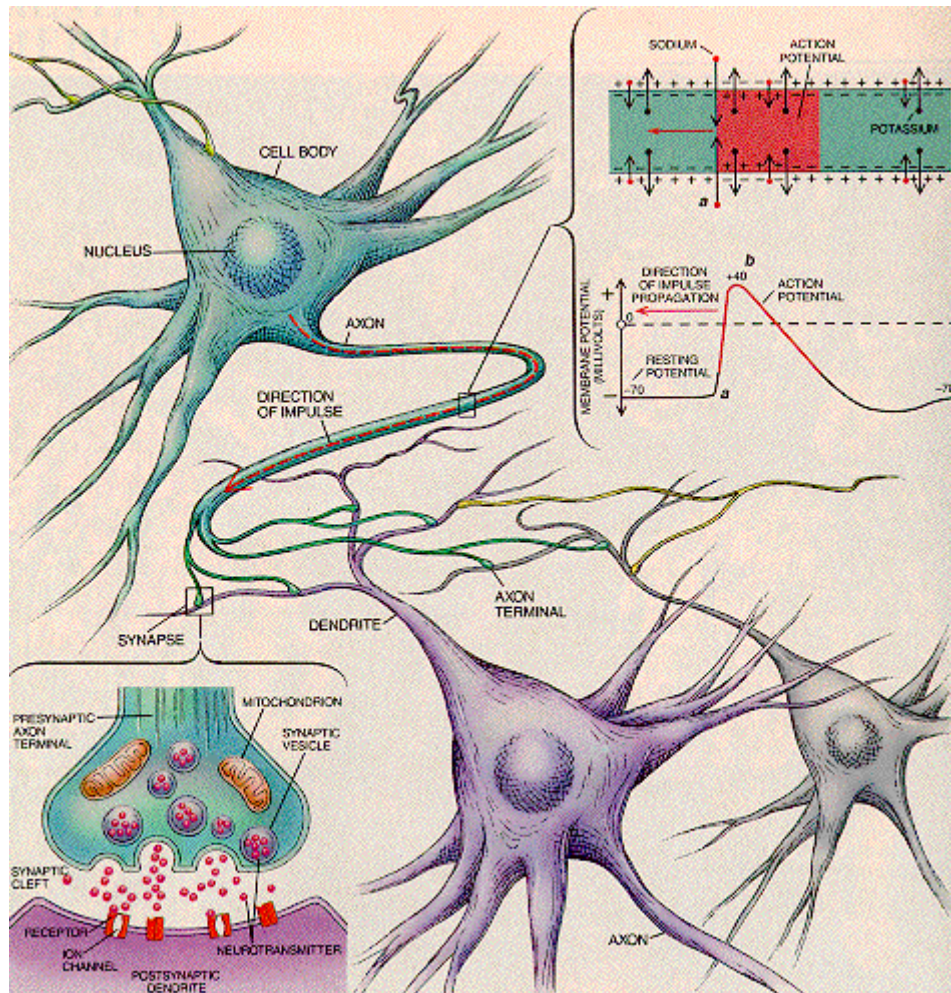


Figure 6 : Modèle de neurone biologique. Notion de connectivité par schéma qui peut montrer l'effet de relais d'un signal transmis d'un neurone à un autre via axone et dendrites.

Un neurone est constitué de trois parties :

2.1.1. Le corps cellulaire

Il contient le noyau du neurone et effectue les transformations biochimiques nécessaires à la synthèse des enzymes et d'autres molécules qui assurent la vie du neurone. Sa forme est pyramidale ou sphérique dans la plupart des cas, Elle dépend souvent de sa position dans le cerveau, ainsi les neurones du néo-cortex ont principalement une forme pyramidale. Ce corps cellulaire fait quelques microns de diamètre.

2.1.2. Les dendrites

Chaque neurone possède une chevelure de dendrites. Celle-ci est constituée de fines extensions tubulaires, de quelques dixièmes de microns de diamètre et d'une longueur de quelques dizaines de microns. Elles se ramifient, ce qui les amène à former une espèce d'arborescence autour du corps cellulaire. Elles sont les récepteurs principaux du neurone pour capter les signaux qui lui parviennent.

2.1.3. L'axone

L'axone est la fibre nerveuse qui sert de transporter les signaux émis par les neurones. Il se distingue des dendrites par sa forme et par les propriétés de sa membrane externe. En effet,

il est généralement plus long que les dendrites et se ramifie à son extrémité, là où il communique avec d'autres neurones, alors que les ramifications des dendrites se produisent plutôt près du corps cellulaire.

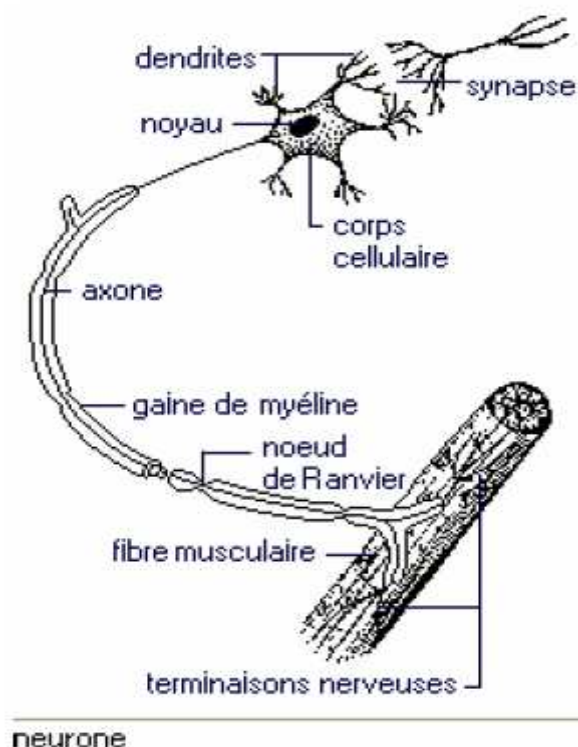


Figure 7 : cellule nerveuse

2.2. La synapse

La synapse est un lieu d'interactions fort complexes où le signal électrique de l'impulsion nerveuse est converti en un signal biochimique. Ce qui nous permet de dire que le rôle des synapses est fondamental pour permettre aux cellules nerveuses de communiquer entre elles.

2.3. Principe de fonctionnement

D'une façon simple, le corps cellulaire traite les impulsions électriques qui lui proviennent des ses dendrites, et qu'il transmet l'impulsion électrique résultante aux neurones auxquels il est connecté par l'intermédiaire de son axone.

Le schéma classique présenté par les biologistes est celui d'un soma effectuant une sommation des influx nerveux transmis par ses dendrites. Si la sommation dépasse un seuil, le neurone répond par un influx nerveux ou potentiel d'action qui se propage le long de son axone. Si la sommation est inférieure à son seuil, le neurone reste inactif. L'influx nerveux qui se propage entre différents neurones est, au niveau de ces neurones, un phénomène électrique.

3. Neurone formel

3.1. Définition

Un neurone formel est un modèle très simplifié du neurone biologique, c'est un automate possédant plusieurs entrées et une seule sortie capable d'effectuer une opération mathématique.

3.2. Modélisation

Chaque neurone formel est défini par les cinq éléments suivants :

- Un ensemble d'influences provenant d'autres neurones par l'entre mise de liens synaptiques pondérés.
- Une étape interne de calcul où ces influences sont transformées en une valeur unique: l'activation pondérée qui décrit l'effet des autres neurones du réseau sur le neurone courant.
- Un seuil d'activation Θ du neurone, valeur de l'activation pondérée au dessous de laquelle le neurone ne répondra pas.
- Une entrée extérieure permettant d'exprimer des informations aux neurones d'entrée du réseau.
- Enfin une fonction qui détermine l'activation réelle du neurone à partir de son activité pondérée, de son seuil et de son entrée extérieure. C'est la fonction d'activation $F(x)$.

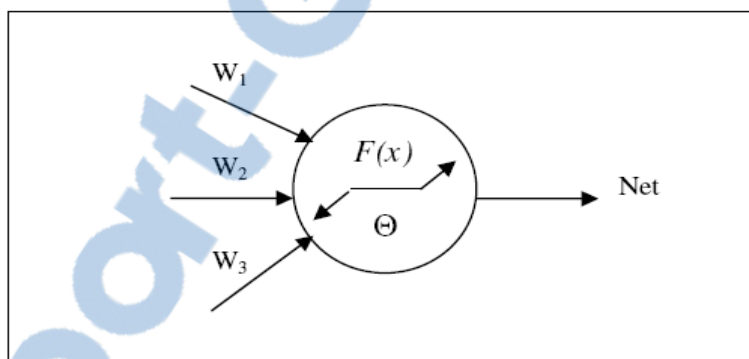


Figure 8 : Modèle général d'un neurone formel

Ce modèle s'exprime mathématiquement par la relation :

$$a_i^{t+1} = f_i(a_i^{t+1}, Net_i^t, \Theta_i, I_i^t)$$

Où :

- a_i^t est l'activation du neurone i au temps t .
- Net_i^t est son activation pondérée au temps t .
- Θ_i est son seuil.
- I_i^t est l'entrée extérieure au temps t .

- f_i est sa fonction d'activation.

En général Net_i^t est calculée par une simple somme pondérée des influences d'entrée du neurone :

$$Net_i^t = \sum_{j \in N} w_{ij} a_j^t$$

Cependant, une fonction plus élaborée peut être utilisée, comme par exemple la fonction de la distance entre l'activation reçue par le neurone sur un lien et le poids de ce lien :

$$Net_i^t = \sqrt{\sum_{j \in N} (w_{ij} - a_j^t)^2}$$

III. Réseaux de Neurones Artificiels « RNA »

1. Définition

Un réseau de neurones est une fonction paramétrée qui est la composition d'opérateurs mathématiques simples appelés neurones formels pour les distinguer des neurones biologiques.

Les réseaux de neurones fonctionnent en répartissant les valeurs des variables dans les neurones. Ces unités sont chargées de combiner entre elles leurs informations pour déterminer la valeur du paramètre de discrimination. La capacité de discrimination du RNA émerge à partir de la connexion de ces unités entre elles.

Chaque neurone reçoit des informations numériques en provenance de neurones voisins ; à chacune de ces valeurs est associée un poids représentatif de la force de la connexion. Chaque neurone effectue localement un calcul dont le résultat est transmis ensuite aux neurones avants.

2. Objectif

L'objectif général d'un RNA est de trouver la configuration des poids de connexion entre neurones pour qu'il associe à chaque configuration d'entrée, une réponse adéquate. L'utilisation d'un RNA se fait en deux temps. Tout d'abord une phase d'apprentissage qui est chargée d'établir des valeurs pour chacune des connexions du réseau, puis une phase d'utilisation proprement dite, où l'on présente au réseau une entrée et où il nous indique en retour sa sortie calculée.

Les réseaux de neurones sont essentiellement utilisés pour faire de la classification. Construit à partir d'exemples de chaque classe qu'il a appris, un réseau de neurones est normalement capable de déterminer à quelle classe appartient un nouvel élément qui lui est soumis.

3. Fonctionnement

Avant de pouvoir utiliser les capacités de classification d'un réseau de neurones, il faut le construire. Ceci se déroule en quatre temps : [Gilles Balmisse, 2002]

- La construction de la structure du réseau (généralement empirique).
- La constitution d'une base de données de vecteurs représentant au mieux le domaine à modéliser. Celle-ci est scindée en deux parties : une partie servant à l'apprentissage du réseau (on parle de base d'apprentissage) et une autre partie au tests de cet apprentissage (on parle de base de test).
- Le paramétrage du réseau par apprentissage. Au cours de l'apprentissage, les vecteurs de données de la base d'apprentissage sont présentés séquentiellement et plusieurs fois au réseau. Un algorithme d'apprentissage ajuste les poids u réseau afin que les vecteurs soient correctement appris. L'apprentissage se termine lorsque l'algorithme atteint un état stable.
- La phase de reconnaissance qui consiste à présenter au réseau chacun des vecteurs de la base de test. La sortie correspondante est calculée en propageant les vecteurs à travers le réseau. La réponse du réseau est lue directement sur les unités de sortie et comparée à la réponse attendue (désirée).

Une fois que le réseau présente des performances acceptables, il peut être utilisé pour répondre au besoin qui a été à l'origine de sa construction.

4. Différents réseaux de neurones

Un réseau de neurones est une composition des fonctions neurones définies précédemment. On distingue généralement deux types de neurones :

- Les neurones cachés : ils sont appelés ainsi parce que leurs sorties ne sont pas des sorties du réseau.
- Les neurones de sortie : leurs sorties sont les sorties du réseau.

L'architecture d'un réseau peut être aussi classée selon son évolution au cours de l'apprentissage et de son utilisation. En fonction de ce critère, les réseaux sont divisés dans les groupes suivants :

- Les réseaux de neurones statiques.
- Les réseaux de neurones dynamiques.

4.1. Les réseaux de neurones statiques

Le réseau a sa structure définie avant l'apprentissage. La quantité de neurones ainsi que sa structure d'interconnexions ne changent pas. Les seuls changements subis par le réseau se

situent au niveau des poids synaptiques, modifiés au cours de l'apprentissage. Ce type de modèle pose une difficulté : la détermination du bon nombre de neurones et interconnexions à utiliser. Un réseau avec peu d'unités et d'interconnexions risque de ne pas pouvoir apprendre à résoudre correctement un problème, et un réseau avec trop d'unités peut avoir des problèmes de convergence, de complexité et de généralisation [JOD 94a, FLE 94]. Il n'existe pas de méthode optimale qui permet de spécifier automatiquement le nombre exact d'unités à employer pour bien résoudre un problème donné. Les réseaux PMC à Rétro Propagation classiques sont des réseaux statiques.

Un réseau statique à une couche de neurones cachés (perceptron multicouche) est représenté sur la Figure 4. Le graphe dont les noeuds sont les neurones, et les arêtes orientées les « Connexions » entre ceux-ci, est acyclique. C'est pourquoi un tel réseau est appelé **réseau non bouclé**.

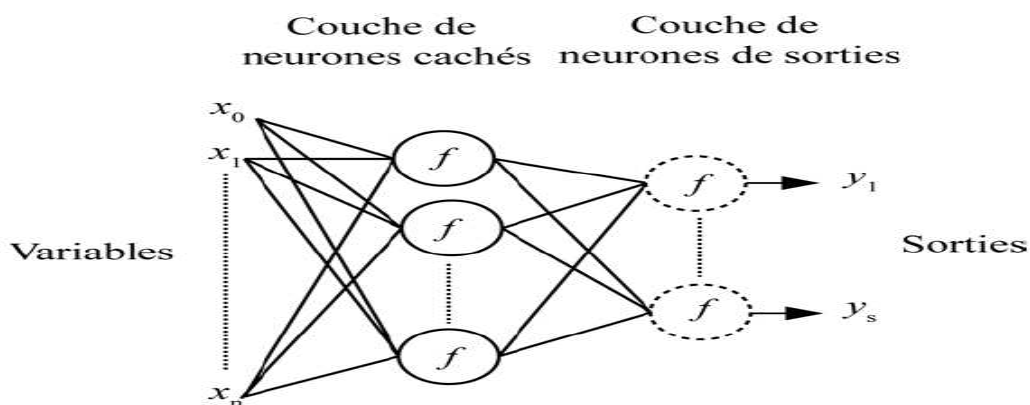


Figure 9 : Un réseau de neurones non bouclé à une couche de neurones cachés (perceptron multicouche)

Dans le cas où les variables d'entrée sont des signaux échantillonnés, les réseaux statiques représentent des filtres numériques transverses non linéaires et non récurrents :

$$y(k) = \varphi_{RN}(\mathbf{x}(k), \dots, \mathbf{x}(k-n))$$

Où :

- $y(k)$ est la sortie du filtre à l'instant discret k .
- $\mathbf{x}(k)$ est le vecteur des variables à l'instant k .
- φ_{RN} est la transformation non linéaire effectuée par le filtre transverse.

4.2. Les réseaux de neurones dynamiques

Les réseaux ayant une structure dynamique sont des réseaux dont le nombre d'unités et d'interconnexions peut varier. Ils sont aussi appelés réseaux ontogéniques [FIE 94b]. Les changements dans la structure du réseau peuvent être de type génératif (incrémental) ou de type destructif (élagage). Le choix entre ces deux méthodes est controversé : faut-il commencer petit et agrandir ou faut-il commencer grand et réduire ensuite ? [ELM 93]. D'un point de vue calculatoire, commencer petit et ajouter des unités et des interconnexions au fur

et à mesure de l'apprentissage est plus performant que faire le travail d'apprentissage sur un grand réseau pour en détruire certaines parties ensuite. Une seule chose semble être acceptée par la plupart des chercheurs : les réseaux ontogéniques sont une des meilleures méthodes dont on dispose pour choisir une bonne architecture pour un réseau et pour bien résoudre un problème donné.

La Figure 5 présente un exemple de réseau de neurones dynamique. Ce réseau possède autant de cycles que de neurones cachés. Ces cycles partent tous d'un neurone caché donné, en passant par l'unique neurone de sortie, puis reviennent au neurone caché de départ avec un retard d'une période d'échantillonnage.

Le graphe de ses connexions, défini dans la section précédente, est cyclique ; c'est pourquoi ce réseau est appelé **réseau bouclé**. Pour qu'il soit causal, il faut que chaque boucle contienne au moins un retard unité. Il réalise alors une ou plusieurs équations récurrentes.

Un réseau bouclé a une sortie, dont les variables d'entrée sont constituées de variables exogènes et de valeurs passées de la sortie, est aussi appelé « filtre récurrent non linéaire ».

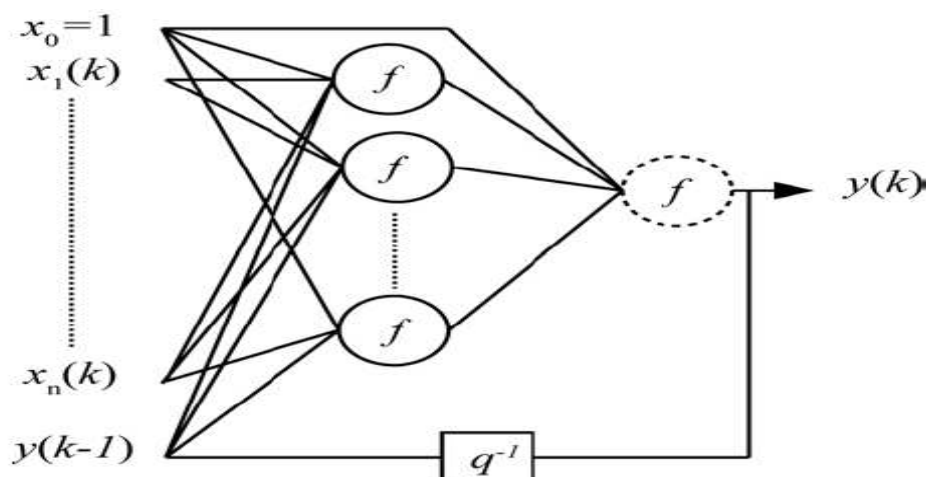


Figure 10 : Réseau de neurones bouclé.

Les travaux de [NERRAND et al. 1993] ont établi que tout réseau de neurones bouclé, aussi complexe soit-il, peut être mis sous une forme particulière, appelée forme canonique, comportant un réseau de neurones non bouclé dont certaines sorties sont ramenées aux entrées par des bouclages de retard unité. Cette forme canonique est par conséquent constituée d'un graphe acyclique, et de connexions à retard unité reliant certaines sorties de ce graphe aux entrées du réseau.

Le dernier point relevant des architectures concerne la modularité des réseaux [AMY 96]. Les réseaux neuronaux peuvent avoir des architectures modulaires : ils peuvent être constitués par des blocs plus ou moins dépendants les uns des autres. On peut avoir des différentes approches pour intégrer et faire coopérer les modules d'un réseau. Une première approche consiste à décomposer le problème et obtenir ainsi des modules spécialisés pour chaque sous problème. Dans cette approche, si l'on prend une tâche de classification en plusieurs classes,

on peut considérer que l'identification de chaque classe va pouvoir être traitée par des modules séparés. L'autre approche, plus difficile à mettre en place, est celle où les différents modules vont coopérer pour aboutir à une solution. Dans cette approche, on ne va pas imposer de tâches particulières à des modules pré spécifiés. La modularité est un problème relatif au choix de l'architecture d'un réseau, mais peut être aussi lié au choix du partitionnement des données. Finalement, la modularité peut devenir un aspect très important à prendre en compte selon la complexité du problème abordé.

5. Propriétés fondamentales des réseaux de neurones

Les réseaux de neurones constituent une famille de fonctions non linéaires paramétrées, que l'on met en oeuvre pour des tâches de modélisation, de prédiction, et de classification.

5.1. L'approximation universelle

La propriété d'approximation universelle des réseaux de neurones a été démontrée dans [CYBENKO 1989] et [FUNAHASHI 1989]:

«Toute fonction bornée suffisamment régulière peut être approchée uniformément, avec une précision arbitraire, dans un domaine fini de l'espace de ses variables, par un réseau de neurones comportant une couche de neurones cachés en nombre fini, possédant tous la même fonction d'activation bornée, et un neurone de sortie linéaire. » [HORNIK et al. 1989], [HORNIK et al. 1990], [HORNIK 1991].

Cette propriété fondamentale est un théorème d'existence. Elle ne simplifie pas pour autant la tâche consistant à identifier les paramètres du réseau. Toutefois, elle permet de fixer l'architecture de réseaux pouvant réaliser l'identification de toute fonction non linéaire.

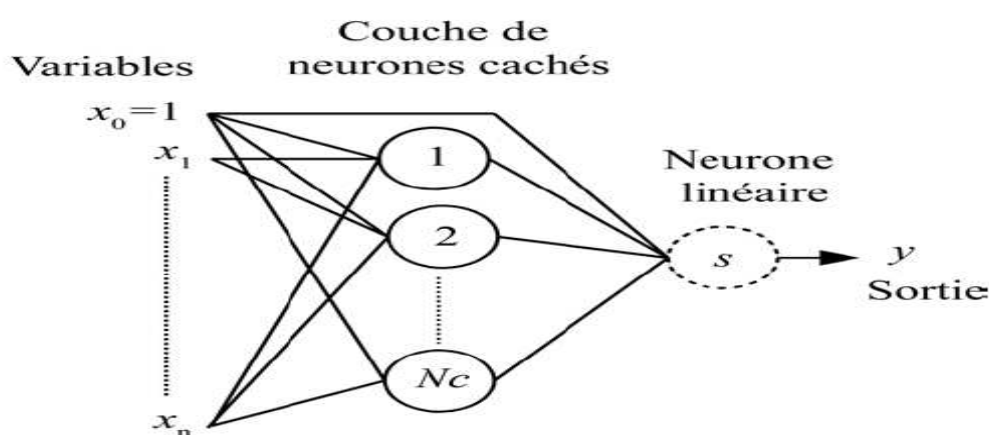


Figure 11 : L'approximateur universel : un réseau de neurones non bouclé à $n+1$ entrées, Nc neurones cachés et un neurone de sortie linéaire

La Figure 6 présente l'approximateur universel : la sortie y de ce réseau s'écrit de la façon suivante :

$$y = w_{s,0} + \sum_{j_2=1}^{N_c} \left[w_{s,j_2} f \left(w_{j_2,0} + \sum_{j_1=1}^n (w_{j_2,j_1} x_{j_1}) \right) \right]$$

Où :

- f est la fonction d'activation sigmoïde des neurones de la couche cachée,
- $w_{s,0}$ est le paramètre reliant l'entrée constante d'indice 0 au neurone de sortie d'indice s ,
- $w_{j_2,0}$ est le paramètre reliant l'entrée constante d'indice 0 au neurone de la couche cachée d'indice j_2 , avec $j_2 \in [1, N_c]$,
- w_{s,j_2} est le paramètre reliant le neurone de la couche cachée d'indice j_2 au neurone de sortie d'indice s ,
- w_{j_2,j_1} est le paramètre reliant la variable d'entrée x_{j_1} ($j_1 \in [1, n]$) au neurone de la couche cachée d'indice j_2 .

Il est important de noter que la fonction réalisée par un tel réseau est une fonction non linéaire de ses variables et de ses paramètres.

5.2. La parcimonie

Par ailleurs, [BARRON 1993] montre que : si l'approximation dépend des paramètres ajustables de manière non linéaire, elle est plus parcimonieuse que si elle dépend linéairement des paramètres. Plus précisément, le nombre de paramètres, pour une précision donnée, croît exponentiellement avec le nombre de variables dans le cas des approximateurs linéaires par rapport à leurs paramètres, alors qu'il croît linéairement avec ce nombre pour les approximateurs non linéaires par rapport à leurs paramètres. La parcimonie est donc d'autant plus avantageuse que le nombre de variables du modèle est grand.

Les réseaux de neurones qui ont été décrits dans la section précédente étant non linéaires par rapport à leurs paramètres, ils sont plus parcimonieux que les approximateurs universels linéaires par rapport à leurs paramètres, tels que les polynômes par exemple.

En raison des propriétés fondamentales mentionnées ci-dessus, les réseaux de neurones sont capables d'intervenir dans la résolution de nombreux problèmes de modélisation et de classification à partir des mesures. Ainsi, il est avantageux de les mettre en œuvre pour toute application nécessitant de trouver, par des méthodes statistiques, une relation non linéaire entre des données numériques.

Les méthodes neuronales ont sans doute des limitations, et leur mise en œuvre nécessite quelques précautions de bon sens qu'on a déjà citées précédemment :

- L'apport des réseaux de neurones réside dans leur capacité à réaliser des approximations de fonctions de régression non linéaires. Avant d'utiliser des réseaux de neurones dans une application, il faut donc s'assurer de la nécessité d'un modèle non linéaire. En effet, la

mise en œuvre d'un modèle linéaire est généralement plus simple que celle d'un réseau de neurones.

- D'autre part, l'utilisation des réseaux de neurones (et plus généralement, des méthodes statistiques) nécessite un échantillon représentatif de la population étudiée.

Mais il est toujours possible que d'autres approximateurs, donnent pour un problème particulier de meilleurs résultats (résultats plus précis avec le même nombre de paramètres ajustables) que les réseaux de neurones. En fonction de la quantité du temps disponible, il est toujours préférable de tester ces approximateurs.

6. Boîtes à outils neuronales

Tout comme pour les traitements statistiques, les données doivent avoir été recueillies sur des cas représentatifs de la population étudiée et la problématique doit être posée en termes clairs.

En ce qui concerne la classification ou la prédiction de valeurs, l'apprentissage des PMC se fait de façon supervisée, la variable de classement ou la variable à prédire doit être connue. Dans le cas de l'estimation de l'âge au décès à partir du squelette, les collections à observer sont celles pour lesquelles on possède cette information. Il existe cependant d'autres problématiques, telles que la catégorisation à laquelle correspond d'autres types de RNA.

En effet, l'alternative à l'architecture en couches du perceptron multicouche est un réseau où les flux d'information ne suivent pas une direction privilégiée. Ce type de réseau est appelé **réseau bouclé**, dont un cas particulier est celui des réseaux totalement connectés. Dans ce cas, on ne cherche plus à connaître des valeurs de « sortie » du réseau, mais à stabiliser globalement les activités individuelles de chacun des neurones. Les réseaux de Hopfield ou encore les cartes auto organisatrices de Kohonen [Kohonen, 1990] en sont des illustrations. Les réseaux de Kohonen, ont permis, par exemple, la reconnaissance des variables significatives dans les séquences génomiques des grandes banques de données internationales. L'élaboration d'un RNA, quelle que soit son architecture, requiert une certaine expérience. C'est la raison pour laquelle la collaboration avec des spécialistes est la démarche la plus souhaitable, que ce soit pour superviser l'utilisation de logiciels simples, tels que MATLAB, STATISTICA, NEURO ONE et NEURO SHELL qui proposent des modules de RNA, ou pour l'élaboration de RNA à partir de logiciels plus sophistiqués. L'application des réseaux de neurones artificiels étant en plein essor dans tous les domaines de la recherche, cette interdisciplinarité est facilement envisageable.

Les réseaux de neurones artificiels constituent donc un outil de traitement des données qui, au même titre que les statistiques traditionnelles, peut s'appliquer dans le domaine de l'analyse des données biologiques et médicales, les relations complexes entre variables ne manquant pas dans ce domaine. Ils ne constituent pas le remède miracle, mais ils sont une alternative

possible aux approches statistiques courantes qui ne sont pas toujours adaptées aux différentes problématiques.

7. Applications des RNA

Les grands domaines d'application des réseaux de neurones découlent naturellement de leur propriété fondamentale :

7.1. La régression non linéaire, ou modélisation de données statiques

Il existe une immense variété de phénomènes statiques qui peuvent être caractérisés par une relation déterministe entre des causes et des effets ; les réseaux de neurones sont de bons candidats pour modéliser de telles relations à partir d'observations expérimentales, sous réserve que celles-ci soient suffisamment nombreuses et représentatives.

7.2. La modélisation de processus dynamiques non linéaires

Modéliser un processus, c'est trouver un ensemble d'équations mathématiques qui décrivent le comportement dynamique du processus, c'est-à-dire l'évolution de ses sorties en fonction de celle de ses entrées ; c'est donc typiquement un problème qui peut être avantageusement résolu par un réseau de neurones, si le phénomène que l'on désire modéliser est non linéaire. La prédiction de séries chronologiques (prédictions financières, prédiction de consommation, etc.) entre dans ce cadre.

7.3. La commande de processus

Commander un processus, c'est imposer à celui-ci un comportement défini à l'avance en fonction des signaux de commande ; l'ensemble {commande, processus} peut donc être considéré comme un système qui réalise une fonction non linéaire qu'un réseau de neurones peut approcher.

7.4. La classification

Si on désire classer des formes en deux catégories, A ou B, en fonction de certaines caractéristiques de ces formes ; on peut définir une fonction ϕ qui vaut +1 pour toutes les formes de la classe A et -1 pour toutes les formes de la classe B. Les réseaux de neurones sont de bons candidats pour réaliser une approximation de cette fonction ϕ , et c'est démontrable que cette approximation constitue une estimation de la probabilité d'appartenance de la forme inconnue à la classe A. Les réseaux de neurones fournissent donc une information très riche, qui est loin d'être une simple réponse binaire. Cette propriété remarquable (que les réseaux de neurones partagent avec d'autres classifieurs) n'est malheureusement pas mise à profit dans la plupart des applications.

IV. Apprentissage des réseaux de neurones

L'apprentissage des réseaux de neurones consiste à estimer les paramètres w du modèle, de telle manière qu'une distance convenablement choisie entre les observations appartenant à un ensemble de données, l'ensemble d'apprentissage, et les prédictions du modèle pour ces mêmes données soit aussi petite que possible.

1. Procédure d'apprentissage d'un RNA

L'apprentissage dans les réseaux de neurones nécessite :

- **Un ensemble d'exemples d'apprentissage** : en effet, les réseaux de neurones sont des fonctions paramétrées, utilisées pour réaliser des modèles statistiques à partir d'exemples (dans le cas de la classification) ou de mesures (dans le cas de la modélisation) ; leurs paramètres sont calculés à partir de ces exemples ou couples {entrée, sortie}.
- **Définition d'une fonction de coût** : elle mesure l'écart entre les sorties du réseau de neurones et les sorties désirées (classification) ou les valeurs mesurées (modélisation) présentes dans l'ensemble d'apprentissage.
- **Un algorithme de minimisation de fonction de coût** : et cela par rapport au paramètres.

2. Position du problème

A partir d'une architecture de réseau de neurones donnée, il est possible d'engendrer une famille de fonctions paramétrées par les valeurs de coefficients du réseau (ou poids synaptiques). L'objectif de la phase d'apprentissage des réseaux de neurones est de trouver parmi toutes ces fonctions, celle qui s'approche le plus possible à la régression (fonction génératrice des exemples).

La fonction génératrice est inconnue, sinon ce n'est pas nécessaire d'utiliser une approximation par réseaux de neurones, seulement les valeurs observées sont connues (qui représentent les valeurs de la régression à laquelle s'ajoute du bruit) pour plusieurs valeurs prises par les entrées (éléments de l'ensemble d'apprentissage). Autrement dit, la fonction de régression doit être trouvée ($E(y|x)$: espérance mathématique des valeurs observées y au point x), et comme le nombre de points est fini, l'approximation doit donc être unique.

Une telle approximation est trouvée à travers la définition d'une fonction de coût qui mesure l'écart entre la sortie du modèle et la sortie désirée. La fonction de coût est une fonction scalaire qui dépend du vecteur de paramètres θ du modèle, et des individus de l'ensemble d'apprentissage. Dans le cas des réseaux de neurones, le vecteur de paramètres est constitué par les poids du réseau. Plus la valeur de la fonction de coût est petite, plus le modèle traite fidèlement les observations utilisées pour l'apprentissage. Et donc, trouver le point dans

l'espace des paramètres, pour lequel la fonction de coût est minimale représente l'objectif majeur des algorithmes d'apprentissage.

3. Fonction de coût

3.1. Présentation

La définition d'une fonction de coût est primordiale, car celle-ci sert à mesurer l'écart entre la sortie du modèle et les mesures faites sur les exemples d'apprentissage. Il existe un grand nombre de fonctions possibles [Bishop 95]. La fonction la plus couramment utilisée, est la fonction dite des "moindres carrés".

3.2. Fonction de coût des moindres carrés

Pour un exemple d'un ensemble d'observations, la fonction de coût des moindres carrés est égale à la somme, sur les neurones de la couche de sortie, des carrés des écarts entre la sortie du modèle (sortie du réseau de neurones) et la sortie désirée (grandeur mesurée). Comme la sortie du réseau de neurones dépend du vecteur de paramètres θ , la fonction de coût en dépend également.

$$J^i(\theta) = \sum_{q=1}^{N_s} (d_q^i - s_q^i)^2 \quad \text{Pour un exemple}$$

$$J^E(\theta) = \frac{1}{N} \cdot \sum_{\substack{i=1 \\ i \in E}}^N J^i(\theta) \quad \text{Sur un ensemble d'exemples}$$

L'apprentissage consiste à minimiser une fonction de coût à l'aide des algorithmes d'optimisation qui vont être décrit par la suite. Si la sortie du modèle est linéaire par rapport aux paramètres, l'apprentissage s'effectue en une seule étape avec la méthode des moindres carrés ordinaire. Les méthodes itératives qui assurent la décroissance de la fonction de coût et convergent vers un minimum, sont utilisées dans le cas contraire (modèles non linéaires).

3.3. Minimisation de la fonction de coût

Le modèle étant non linéaire en ses paramètres, la fonction de coût n'est pas quadratique. La méthode des moindres carrés n'est donc pas applicable. En conséquence, on a recours à des méthodes itératives d'optimisation de la fonction de coût.

Toutes les méthodes d'optimisation utilisent le gradient de la fonction de coût. La première étape de l'apprentissage d'un réseau de neurones consiste donc à calculer le gradient de la fonction de coût, à l'aide de l'algorithme de rétropropagation [RUMELHART et al. 1986].

Une fois le gradient calculé, un algorithme itératif de modification des paramètres est mis en oeuvre. Parmi ces derniers, on distingue les méthodes itératives du premier ordre et les méthodes du second ordre.

Les méthodes du premier ordre modifient itérativement les paramètres de manière proportionnelle au gradient de la fonction de coût, avec un coefficient de proportionnalité fixe ou variable au cours du déroulement de l'optimisation. En d'autres termes, l'extrémité du vecteur des paramètres se déplace, à chaque itération, dans la direction du gradient de la fonction de coût.

Dans les méthodes du second ordre, la direction de déplacement du vecteur des paramètres est obtenue par une transformation linéaire du gradient de la fonction de coût, transformation qui fait intervenir la matrice des dérivées secondes de la fonction de coût par rapport aux paramètres (matrice Hessienne). Ces méthodes sont beaucoup plus efficaces que les méthodes du premier ordre. Le choix entre les diverses méthodes du second ordre dépend notamment du nombre de paramètres des modèles étudiés. Cette méthode est avantageuse car il n'est pas nécessaire de calculer les dérivées secondes de la fonction de coût.

4. Le calcul du gradient

Sauf dans le cas d'un modèle linéaire par rapport aux paramètres, la minimisation de la fonction de coût s'effectue de manière itérative, par une descente de gradient.

Il faut donc disposer d'une méthode de calcul du gradient de la fonction de coût par rapport aux poids du réseau. L'essor récent des réseaux de neurones tient en partie à la mise au point, par [Rumelhart 86], de l'algorithme de rétro propagation.

Cette méthode permet de calculer de manière particulièrement simple et rapide, en tout point de l'espace des entrées, le vecteur de taille q , nombre de paramètres ajustables du réseau.

5. L'algorithme d'optimisation (apprentissage)

L'algorithme d'optimisation permet d'ajuster de manière itérative les poids du modèle de façon à converger vers un minimum de la fonction de coût. Pour ce faire, on calcule la modification des coefficients à l'étape k par la formule suivante :

$$\theta_k = \theta_{k-1} + \mu_{k-1} d_{k-1}$$

Où μ_{k-1} et d_{k-1} représentent respectivement le pas et la direction de descente.

Ainsi, on part d'une initialisation aléatoire des poids, et l'on poursuit de déroulement de l'algorithme jusqu'à satisfaction d'un critère d'arrêt, portant par exemple sur la norme du vecteur dérivée de la fonction de coût par rapport aux coefficients ou sur le nombre d'itérations. Sauf dans le cas linéaire, pour lequel la fonction de coût est quadratique, la fonction de coût possède plusieurs minima vers lesquels peut converger l'apprentissage, suivant l'initialisation des poids.

Les algorithmes utilisés se différencient par le choix du pas et de la direction de descente :

- Le pas peut être soit constant, soit ajusté en cours d'apprentissage, par exemple par la méthode de Nash [Nash 90] ou celle de Wolfe et Powell ([Wolfe 69] et [Powell 76]),

- Dans un algorithme du premier ordre, la direction de descente est calculée à partir de la dérivée première de la fonction de coût par rapport aux coefficients,
- Dans un algorithme du second ordre, on utilise le Hessien (c'est-à-dire la matrice des dérivées secondes de la fonction de coût par rapport aux coefficients), ou une approximation de celui-ci, pour déterminer la direction de descente.

Dans le cas général, il est fortement conseillé d'utiliser des méthodes du second ordre, beaucoup plus efficaces que les méthodes du premier ordre. C'est la raison pour laquelle ces méthodes sont fréquemment utilisées, telles que la méthode de quasi-Newton, [Press 92], ainsi que l'algorithme de Levenberg-Marquardt ([Levenberg 44] et [Marquardt 63] et [BENDER 1996]), cette méthode est avantageuse car il n'est pas nécessaire de calculer les dérivées secondes de la fonction de coût.

V. Les différents algorithmes d'apprentissage

Il faut choisir l'algorithme d'optimisation à partir d'une fonction de coût et d'une base d'apprentissage, cet algorithme va permettre d'estimer le vecteur des paramètres pour lequel la fonction de coût choisie est minimale. Il existe de nombreux algorithmes, on va citer par la suite quelques uns qui sont fréquemment mis en œuvre pour l'apprentissage des réseaux de neurones.

1. Modèles linéaires par rapport aux paramètres

Un modèle linéaire par rapport aux paramètres est défini par l'équation :

$$Y = X \theta + \omega$$

Avec :

$$Y = \begin{bmatrix} y^1 \\ \vdots \\ y^N \end{bmatrix} : \text{Vecteur de sortie de dimension } N \text{ (} N = \text{nombre d'exemples)}$$

$$X = \begin{bmatrix} x_1^1 & \cdots & x_P^1 \\ \vdots & & \vdots \\ x_1^N & \cdots & x_P^N \end{bmatrix} : \text{Matrice des entrées, de dimension } N \times P \text{ (} P = \text{nombre de colonnes correspondant aux } P \text{ variables descriptives du modèle, et } N \text{ lignes représentant les } N \text{ exemples)}$$

- θ : vecteur de dimension P des paramètres inconnus du modèle (dans ce cas, P = nombre de descripteurs).

- ω : vecteur du bruit, centré, non corrélé, de dimension N, normalement distribué (de moyenne nulle et de variance σ).

Avec un tel modèle, l'estimation des moindres carrés des paramètres () est la solution de l'équation normale :

$$X^T X \hat{\theta} = X^T Y$$

Soit
$$\hat{\theta} = [X^T X]^{-1} X^T Y, \text{ si } \det(X^T X) \neq 0$$

La solution de cette équation s'obtient par diverses méthodes telles que les méthodes d'orthogonalisation, la décomposition de Cholesky,... [Antoniadis 92 et Press 92].

2. Modèles non linéaires par rapport aux paramètres

Un modèle non linéaire par rapport aux paramètres se définit par l'équation suivante :

$$Y = f(X, \theta) + \omega$$

Avec f : fonction de régression.

Si le modèle est non linéaire par rapport aux paramètres, les méthodes de résolution citées dans le paragraphe ne sont plus utiles ; et il faut onc faire appel à des méthodes itératives pour trouver une estimation des paramètres. Ces dernières, sont assez simples à mettre en œuvre et elles sont applicables à toutes les fonctions de coûts dérivables par rapport à θ .

Ces méthodes reposent sur le principe de se positionner en un point, trouver une direction de la descente du coût dans l'espace des paramètres θ , et puis de se déplacer d'un pas en suivant cette direction. Et ça se répète jusqu'à satisfaction d'un critère d'arrêt fixé à l'avance.

A l'itération k, la modification du vecteur de paramètres est donnée par :

$$\theta_k = \theta_{k-1} + \mu_{k-1} \cdot d_{k-1}$$

Avec d_{k-1} : direction de descente, dépendant des θ_{k-1}

μ_{k-1} : le pas

Les méthodes d'optimisation non linéaires se différencient dans le choix du pas et de la direction de la descente. Pour cela elles utilisent :

- Hessien : matrice des dérivées secondes de la fonction de coût par rapport à θ .
- Gradient : vecteur des dérivées premières de la fonction de coût par rapport à θ .

Pour le calcul du gradient dans le cas des réseaux de neurones, l'algorithme de rétropropagation de l'erreur est utilisé [Rumelhart 86]. Au moment où l'approximation du Hessien est utilisée pour les méthodes de quasi-Newton et Levenberg-Marquardt [Bishop 95].

2.1. Méthode du gradient à pas constant

Simple à mettre en œuvre, elle se base sur le calcul du gradient qui donne la direction de la descente. Le pas μ est constant. La modification des paramètres à l'itération k est :

$$\theta_k = \theta_{k-1} - \mu \cdot \nabla J_{k-1}$$

Avec
$$d_{k-1} = -\nabla J_{k-1}$$

$$\mu_{k-1} = \mu \text{ (constant)}$$

La méthode du gradient à pas constant est facile à utiliser et très efficace. Mais le gradient tend vers zéro quand on s'approche du minimum, et donc la vitesse de convergence diminue fortement.

2.2. Méthode du gradient à pas variable

Etant donnée une direction de descente, il est possible d'asservir le pas de telle façon que la fonction de coût diminue à chaque modification des paramètres. Cette méthode est définie par :

$$g(\mu) = J(\theta_{k-1} + \mu \cdot d_{k-1})$$

A partir de la fonction g , il faut chercher une valeur moyenne du pas. Les méthodes les plus efficaces sont :

- a. Celles dites de dichotomie malgré qu'elles requièrent trop de calculs.
- b. Les méthodes de minimisation de Nash [Nash 90] et les méthodes de Wolfe et Powell [Wolfe 69 et Powell 76] permettent de trouver un pas convenable à partir d'un faible nombre d'évaluations de la fonction de coût. Ainsi qu'elles sont considérées les méthodes les plus économiques.

2.3. Méthode de Newton

Elle utilise la dérivée seconde (courbure) de la fonction de coût pour atteindre le minimum rapidement. La modification des paramètres se fait à travers les formules suivantes :

$$\theta_k = \theta_{k-1} - H_{k-1}^{-1} \nabla J_{k-1}$$

Avec
$$d_{k-1} = -H_{k-1}^{-1} \nabla J_{k-1}$$

$$\mu_{k-1} = \mu \text{ (constant)}$$

Dans ce cas, le pas est constant et égal à un. La direction de descente est fonction du Hessian et du Gradient.

Si la fonction de coût est quadratique, l'algorithme atteint la solution en une seule itération. Si ce n'est pas le cas, cette méthode est efficace au voisinage d'un minimum. Néanmoins, le Hessian doit être défini positif pour que la méthode puisse converger vers le minimum. En

pratique, cette condition n'est pas toujours vérifiée, et par la suite la méthode peut ne pas converger. La méthode de Newton n'est pas trop employée à cause des calculs compliqués du Hessien, d'où l'apparition des méthodes économiques dites de quasi Newton.

2.4. Méthode de quasi-Newton

La méthode de quasi-Newton se base sur l'approximation de l'inverse du Hessien, par une matrice positive traitée à chaque itération. La suite des matrices d'approximation est construite de manière à converger vers l'inverse du Hessien lorsque la fonction de coût est une quadrique. La modification des différents paramètres est donnée par :

$$\theta_k = \theta_{k-1} - \mu_{k-1} \cdot M_{k-1} \nabla J_{k-1}$$

$$\text{Avec : } d_{k-1} = -M_{k-1} \nabla J_{k-1}$$

μ_{k-1} est évalué avec une méthode de minimisation unidimensionnelle.

A la première itération, la matrice d'approximation est égale à la matrice identité. Il existe une large variété des méthodes de quasi-Newton [Minoux 83], mais la plus fréquemment utilisée est celle appelée la méthode BFGS, développée par Broyden [Broyden 70], Fletcher [Fletcher 70], Goldfarb [Goldfarb 70] et Shanno [Shanno 70]. La méthode BFGS est caractérisée par sa grande vitesse de convergence par rapport à la méthode de gradient, en plus de son insensibilité au choix du pas calculé avec la méthode de Nash.

2.5. Méthode de Levenberg-Marquardt

L'algorithme de Levenberg-Marquardt permet d'obtenir une solution numérique au problème de minimisation d'une fonction généralement non linéaire dépendant de plusieurs paramètres.

L'algorithme de Levenberg-Marquardt est particulièrement astucieux car il s'adapte lui-même à la forme de fonction de coût, et cela en effectuant un compromis entre la direction du gradient et la direction donnée par la méthode de Newton.

Levenberg et Marquardt ont proposé une méthode efficace pour passer continûment du schéma d'inversion du Hessien à celui des plus fortes pentes. Ce dernier sera utilisé loin du minimum et on tend à lui substituer le schéma d'inversion du Hessien au fur et à mesure que l'on approche du minimum. Cette méthode a fait ses preuves et fonctionne pour des modèles et des domaines de la physique variés, si bien qu'elle constitue le standard pour résoudre les problèmes d'ajustement aux moindres carrés non linéaires [Malik Ndoeye, 2008].

VI. Amélioration de l'apprentissage

L'amélioration des performances d'un système d'apprentissage par réseaux de neurones artificiels, nécessite une meilleure compréhension des relations entre structure (topologie du réseau et ses connexions) et dynamique (propagation des activités).

Plusieurs méthodes d'optimisation existent, les plus couramment utilisées sont :

- Heuristiques par essais erreurs.
- Algorithmes génétiques / programmation évolutionnaire.

Pour réaliser une amélioration d'apprentissage, il est nécessaire de respecter certains critères comme:

- Une bonne préparation des données d'apprentissage concernant le choix des paramètres qui composent le vecteur d'entrée (normalisation). Et cela afin de réduire le temps d'apprentissage ainsi que l'espace de stockage.
- Le bon choix de l'architecture optimale du réseau de neurones.
- La minimisation de l'erreur quadratique.
- La minimisation du temps d'apprentissage.

1. Normalisation

D'après A.Varfis et C.Versino en 1990, il est nécessaire d'effectuer des prétraitements sur les données dans le but de supprimer toute redondance, ce qui permet de réduire la complexité du réseau et accélérer sa convergence.

La normalisation des données est considérée comme le pré-traitement le plus important lorsqu'on utilise les réseaux de neurones. Dans la littérature, plusieurs types de normalisation existent.

Kouam en 1993 a mis en valeur l'influence que peut avoir la normalisation sur la qualité de la prévision, il propose trois types de normalisation : soit la normalisation de la base d'apprentissage, soit la normalisation des valeurs passées de la série, ou la normalisation des valeurs utilisées par le modèle pour calculer la prévision.

2. Optimisation de l'architecture

L'optimisation de l'architecture a pour objectif de réduire dans la mesure de possible le nombre de synapses (poids) et de neurones afin de réduire la complexité du réseau et améliorer le temps de calcul, et maintenir les capacités de généralisation (apprentissage). En effet si par exemple un nombre trop faible de neurones est utilisé, des mauvaises performances en apprentissage et en généralisation sont obtenues, et d'un autre coté, c'est-à-

dire si un très grand nombre de neurones est utilisé, il se provoquera une limite de la capacité de généralisation, ce qui est appelé "apprentissage par cœur". Concernant l'optimisation de l'architecture du réseau, deux principales approches ont été proposées dans la littérature à savoir l'approche par sélection (approche destructive) et l'approche incrémentale (approche constructive).

2.1. L'approche par sélection

L'approche par sélection ou approche destructive consiste à commencer par la construction d'un réseau complexe qui contient un grand nombre de neurones, puis cette approche consiste à essayer de réduire le nombre de neurones inutiles et supprimer les connexions redondantes pendant ou à la fin de l'apprentissage.

- Optimal Cell Damage (OCD) : Cette méthode qui est directement inspirée de l'Optimal Brain Damage (OBD) proposé par Le Cun en 1990, elle consiste à supprimer des cellules durant l'apprentissage. Une cellule est supprimée si sa sensibilité est faible.

La sensibilité d'une cellule est définie par : $S_{Ci} = \prod_{j=1}^n S_i(W_j)$; où :

- C_i représente la cellule i du réseau de neurones,
- W_j représente la connexion sortante j de la cellule C_i et
- $S_i(W_j)$ représente la sensibilité de la connexion W_j . Cette sensibilité est définie par :

$$S_i(W_j) = \frac{1}{2} \frac{\partial^2 MSE}{\partial W_j^2} W_j^2 \text{ avec MSE est la moyenne des carrées des erreurs.}$$

En 1996, Cibas propose alors la démarche suivante : D'abord faire converger le réseau de neurones, puis calculer la sensibilité pour chaque entrée du réseau, ensuite classer les entrées par ordre croissant de sensibilité, enfin supprimer les entrées pour lesquelles la sensibilité est inférieure à un seuil s ; le processus sera répété jusqu'à aboutir à une architecture optimale [H. El Ayeche et A. Trabelsi].

2.2. L'approche incrémentale

Dans l'approche incrémentale ou approche constructive, et à l'inverse de l'approche précédente, on commence par un réseau le plus simple possible, puis on y ajoute des neurones ou des couches, jusqu'à avoir une architecture optimale.

La méthode de Fahlman en 1989 qui commence par un réseau initialisé à un seul neurone. Une fois le neurone est bien entraîné, une nouvelle unité sera ajoutée dans le réseau et ainsi de suite...L'inconvénient de l'approche de Fahlman est qu'elle induit à un réseau dont le nombre de couches est élevé [H. El Ayeche et A. Trabelsi].

L'optimisation de l'architecture des réseaux est nécessaire pour que celle-ci soit représentative de la complexité du modèle à identifier. Pour cela une démarche originale est retenue. Il s'agit d'un algorithme de descente du gradient modifié.

Ce nouvel algorithme part du principe du développement en série de Taylor. Soient x et x_0 deux réels. Toute fonction C^n de x peut s'écrire de la manière suivante au voisinage de x_0 :

$$f(x) = f(x_0) + \sum_{i=1}^n (x - x_0)^i \cdot \left. \frac{d^i f}{dx^i} \right|_{x=x_0}$$

Si f est le nombre de neurones sur la couche interne correspondant au réseau d'erreur maximale x , x_0 étant l'erreur du réseau courant (E_c) et $f(x_0)$ l'architecture du réseau courant (A_c), on peut ainsi utiliser cette relation pour trouver l'architecture désirée (A_d) en fonction de l'erreur maximale désirée (E_d) :

$$A_d = \text{round} \left(A_c + \sum_{i=1}^n (E_d - E_c)^i \cdot \left. \frac{d^i f}{dx^i} \right|_{x=x_0} \right)$$

Le calcul de la dérivée i ème de f par rapport à x nécessite d'avoir au moins $i+1$ échantillons de f . Les méthodes actuelles sont les suivantes :

- Pour la descente du gradient on n'utilise que la dérivée première aux deux derniers points de l'optimisation.
- Pour l'algorithme de newton, on utilise la dérivée première et la dérivée seconde (aux trois derniers points d'optimisation).

Ces principes de calcul sont relativement sensibles à la présence des minimas locaux. C'est pourquoi est proposé un algorithme adaptatif en fonction du nombre de points d'optimisation. En effet, au fur et à mesure que l'algorithme avance, l'échantillon de points s'enrichit. A l'étape $n+1$, on dispose de $n+1$ points et donc on peut utiliser la forme de Taylor à l'ordre n . Ceci permet d'affiner l'optimisation au cours du temps. A noter qu'on interdit le test multiple d'une architecture donnée, afin de gagner du temps de calcul.

Lorsque le processus mène à un réseau d'erreur inférieure à celle spécifiée, on l'arrête et on conserve ce réseau final. Il arrive toutefois que les réseaux ne puissent pas converger vers la solution, par manque de points d'apprentissage. Dans ce cas, on stoppe l'optimisation d'architecture du réseau lorsque les trois dernières architectures testées apportent une dégradation de performance. L'opérateur doit donc enrichir la base de données avant de relancer le processus s'il souhaite conserver ses exigences en terme d'erreur maximale du modèle [F. LAVERGNE].

VII. Conclusion

Ce chapitre a permis de rappeler les propriétés principales des réseaux de neurones, ainsi que les différents algorithmes d'apprentissage automatique.

Le grand avantage des réseaux de neurones réside dans leur capacité d'apprentissage automatique, ce qui permet de résoudre des problèmes sans nécessiter l'écriture des règles complexes, tout en étant tolérant aux erreurs.

Cependant, ce sont des véritables boîtes noires qui ne permettent pas d'interpréter les modèles construits. En cas d'erreurs du système, il est quasiment impossible d'en déterminer la cause.

Aussi un aperçu sur l'amélioration de l'apprentissage en utilisant les réseaux de neurones en ce qui concerne la minimisation de l'erreur ainsi que l'optimisation de l'architecture.

Dans le chapitre 3, nous présentons le domaine d'application de notre travail à savoir la cardiologie.

Chapitre 3 : Généralités en cardiologie

I. Introduction

Dans ce chapitre est présenté le fonctionnement général du système cardiovasculaire, puis, de manière plus détaillée, le principe de l'électrocardiogramme (ECG). Cette présentation se limite au strict nécessaire pour une bonne compréhension du mémoire ; le lecteur intéressé par une approche médicale rigoureuse pourra se reporter aux nombreux ouvrages médicaux disponibles sur le sujet, comme par exemple « Le coeur en action » [DeBakey, 1997].

II. Le système cardiovasculaire

Le système cardiovasculaire assure la circulation sanguine et sa régulation dans l'organisme et permet ainsi de transporter vers les cellules l'oxygène et les substances nutritives et d'emmener les déchets et l'activité cellulaire vers les émonctoires ou vers les organes de détoxication. Il comprend :

- Une pompe ; le cœur,
- Des tuyaux ; artères, veines et capillaires.

L'écoulement du sang se fait à travers deux réseaux vasculaires :

- Un circuit à haute pression, le réseau artériel,
- Un circuit à basse pression, le réseau veineux.

Dans la grande circulation, le système artériel assure le transport du sang oxygéné vers les organes, le système veineux le retour du sang pauvre en oxygène (Figure 1) [CHIKH 2005].

III. Anatomie du cœur

Le coeur humain est un muscle creux, ayant la forme d'une pyramide triangulaire. Il est anatomiquement situé dans la cavité thoracique entre les deux poumons, dans un espace appelé médiastin (Figure 3).

Le poids du coeur humain adulte normal peut aller jusqu'à 340 grammes, et un peu moins chez la femme. C'est un organe comportant une couche externe (péricarde), une couche musculaire (myocarde) et une couche interne (endocarde).

Le coeur propulse le sang grâce aux contractions de son myocarde. Une épaisse cloison le divise en deux moitiés (coeur gauche/coeur droit), et chacune d'elles comporte deux cavités : l'oreillette et le ventricule. À chaque battement, le myocarde suit la même séquence de mouvement : le sang pauvre en oxygène arrive au coeur par la veine cave. Il y entre par l'oreillette droite, et en est chassé par sa contraction appelée systole auriculaire qui le déplace

dans le ventricule droit. La systole ventriculaire (contraction des ventricules) propulse à son tour le sang du ventricule droit vers les poumons où il va se charger en oxygène. De retour au coeur par les veines pulmonaires, le sang s'accumule dans l'oreillette gauche puis, lors de la systole auriculaire, passe dans le ventricule gauche qui lors de la systole ventriculaire l'envoie vers les organes par l'artère aorte (Figure 2).

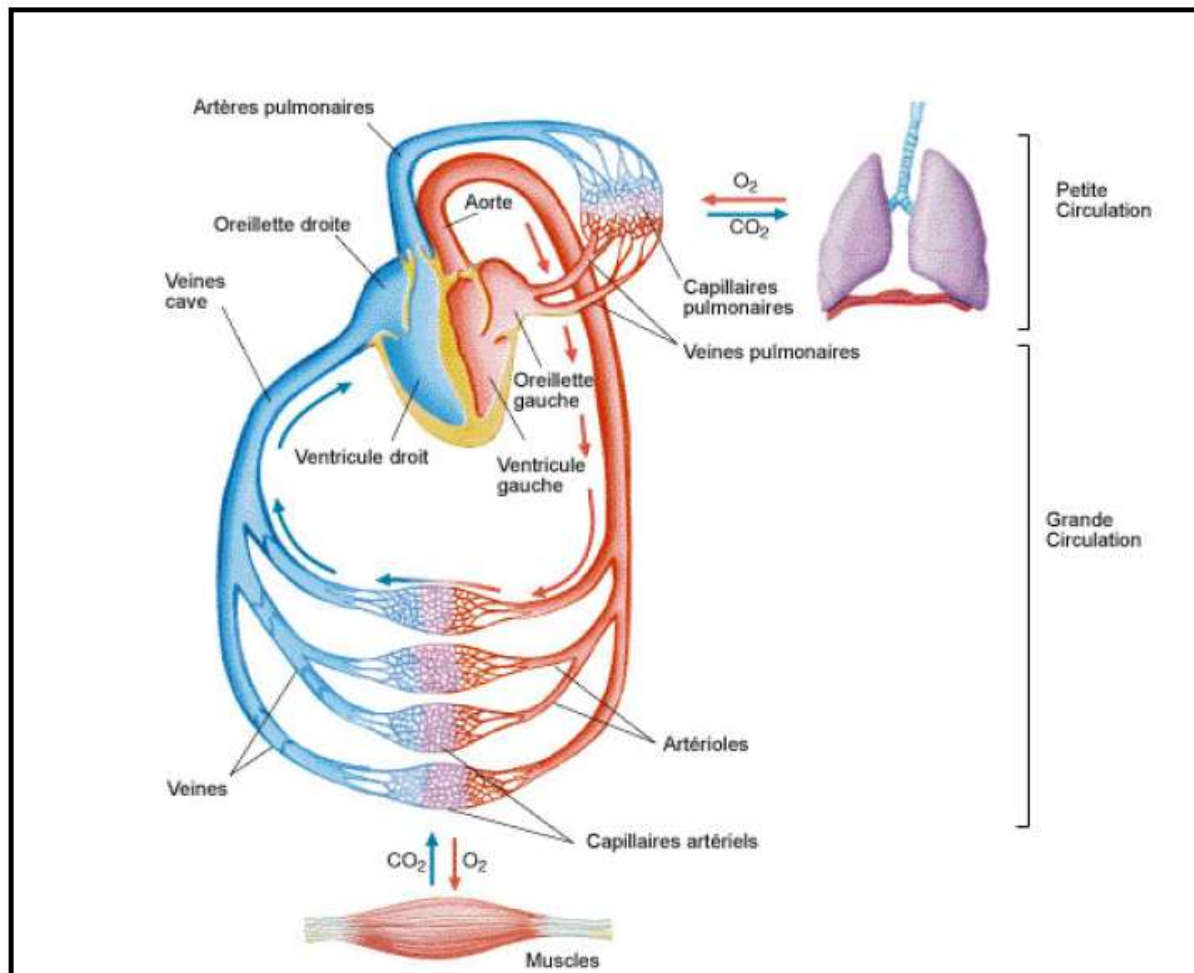


Figure 12: Circulation artérielle et veineuse

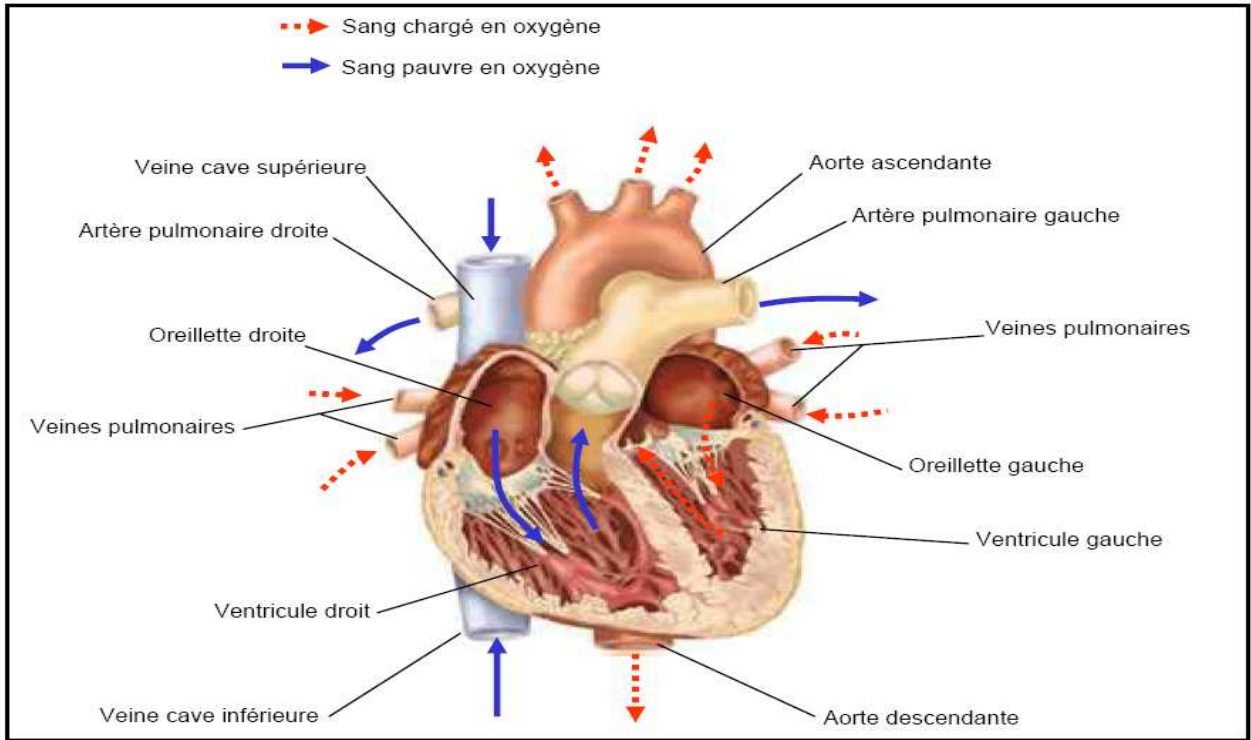
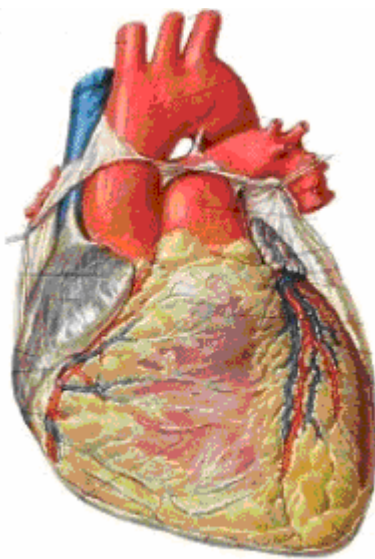
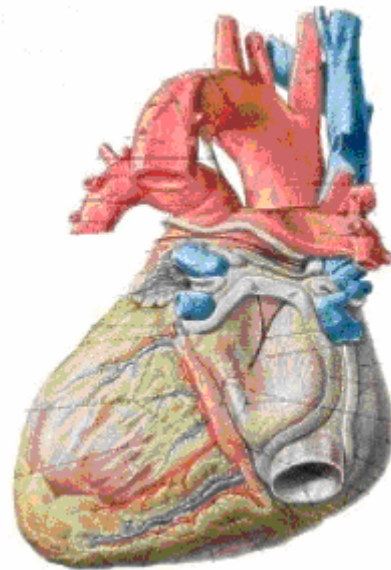


Figure 13 : Anatomie du cœur [Ast 00]



vue ventrale du coeur



Vue dorsale du coeur

Figure 14: vue générale du coeur

1. *Activité électrique du cœur*

Le cœur est animé de contractions qui se produisent à une fréquence variable, décomptées en battements par minute. La fréquence cardiaque moyenne est de 70 à 75 battements par minute, chaque contraction permettant l'éjection d'un volume de 60 cm³ de sang.

Le cœur possède un système nerveux autonome par rapport au reste de l'organisme (propre tissu interne de conduction électrique).



Le courant de contraction prend naissance dans le noeud sinusal appelé aussi "pace maker" (l'appareil du même nom joue le même rôle d'initiation du courant de contraction). En fait, une partie des cellules du coeur est capable de s'auto exciter pour provoquer la naissance d'un courant électrique, qui va se propager dans les oreillettes. Puis le courant atteint un relais (noeud auriculo-ventriculaire), il emprunte ensuite un réseau principal (faisceau de His) et se propage tout autour des ventricules, provoquant leurs contractions et donc l'éjection sanguine (figure 4).

Ce système nerveux autonome de stimulation du coeur peut être influencé par des stimulations nerveuses lui parvenant de son système nerveux périphérique par l'intermédiaire d'hormones (comme l'adrénaline pour le système nerveux sympathique) qui augmentent sa fréquence et sa force de contraction (lors d'un effort sportif le coeur va augmenter son débit sanguin, c'est-à-dire le volume total de sang éjecté dans l'aorte ; le coeur peut débiter jusqu'à cinq fois plus de sang qu'au repos). D'autres stimulations provenant du système nerveux parasympathique, par l'intermédiaire d'autres hormones (comme l'acétylcholine) au contraire vont exercer une influence modératrice sur le "travail" du coeur (ralentissement du rythme et donc de la fréquence de contraction).

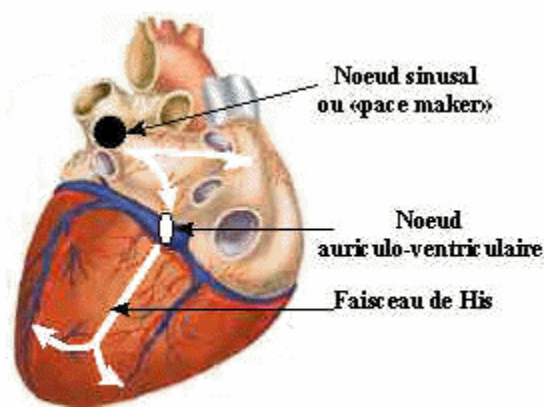


Figure 15 : Les noeuds et faisceau de His

2. *Activité mécanique du coeur*

Le cycle de la circulation sanguine se répète constamment et se divise en deux périodes : la systole et la diastole. La systole est la période correspondant à l'éjection du sang dans la grande et petite circulation. Elle se décompose en trois phases : la systole auriculaire, la contraction ventriculaire isovolumique et la systole ventriculaire. La diastole est la phase de relaxation du coeur, pendant laquelle il se remplit de sang. Cette période est composée de deux phases : la relaxation ventriculaire isométrique et la phase de repos. La succession des différentes phases est détaillée sur la figure ci-dessous :

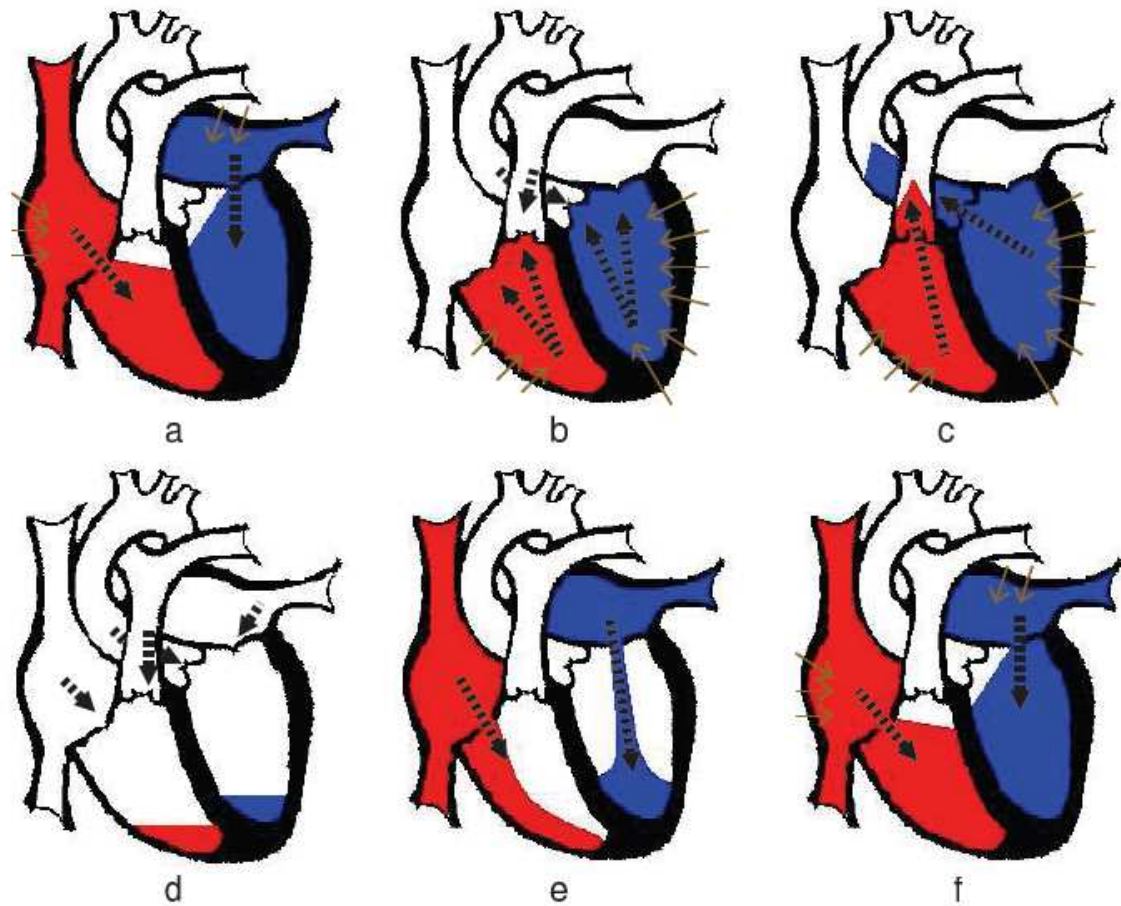


Figure 16 : Cycle mécanique cardiaque. « a) systole auriculaire, b) contraction isovolumique des ventricules, c) systole ventriculaire, d) relaxation des ventricules, e) repos, f) nouvelle systole auriculaire. Les flèches pleines indiquent la contraction musculaire et les flèches en pointillées, la direction de la pression sanguine» .

- **(a) et (f)** La systole auriculaire est la contraction des oreillettes lorsque celles-ci sont remplies de sang. La pression exercée par le muscle auriculaire ferme les orifices des veines caves et pulmonaires et provoque le versement du sang auriculaire dans les ventricules (Figure 5) par les valves tricuspide et mitrale. Cependant, la majorité du remplissage des ventricules survient passivement pendant la diastole.
- **(b)** La contraction ventriculaire isovolumique commence lorsque les cellules musculaires du myocarde ventriculaire se contractent. La pression ferme les valves tricuspide et mitrale et les valvules sigmoïdes restent fermées pendant quelques dizaines de centièmes de secondes. Comme il n'y a pas d'éjection ventriculaire, la pression augmente fortement dans les ventricules.
- **(c)** La systole ventriculaire commence lorsque les pressions dans les ventricules dépassent les pressions dans l'artère pulmonaire et l'aorte. Les valvules sigmoïdes s'ouvrent et l'éjection commence. Le sang oxygéné emprunte la crosse aortique et le sang désoxygéné le tronc pulmonaire. La pression aortique atteint un pic (pression artérielle systolique) puis redescend jusqu'à la fin de la systole.

- **(d)** La relaxation ventriculaire fait suite à la systole. Les ventricules se relâchent, la pression chute jusqu'à être inférieure à celle exercée dans l'aorte et l'artère pulmonaire. En conséquence, les valvules sigmoïdes se ferment et, lorsque la pression devient inférieure à celle des oreillettes, les valves tricuspide et mitrale s'ouvrent.
- **(e)** La phase de repos est celle pendant laquelle le sang des veines caves et pulmonaires s'écoule librement dans les ventricules via les oreillettes.

IV. L'électrocardiogramme (ECG)

L'électrocardiographie est une technique relativement peu coûteuse, qui permet de surveiller l'appareil cardiovasculaire en effectuant un examen indolore et sans aucun risque.

1. Historique

Vers 1880, E Marey [Marey, 1876] et Augustus Waller [Waller, 1893] avaient montré que l'activité électrique du cœur, découverte quelques années plus tôt, pouvait être suivie à partir de la peau ; et vers 1890, Willem Einthoven a réalisé le premier enregistrement cardiographique [Einthoven, 1941]. Le courant mesuré par des électrodes sur le torse du patient mettait en mouvement un mince fil d'argent tendu entre les pôles d'un gros aimant ; ces déflexions étaient enregistrées sur du papier photographique qui se déroulait face à un rayon lumineux.

« Maintenant, nous pouvons enregistrer l'activité électrique du cœur anormale et la comparer à l'activité normale » (Einthoven). Ainsi naît l'électrocardiogramme à l'aube du XX-ième siècle.

2. Principe

Le principe de l'enregistrement moderne est, -à peu de près- le même que celui qui était proposé par Einthoven : grâce à deux électrodes collées à la surface de la peau, on enregistre la différence de potentiel entre deux points diamétralement opposés par rapport au cœur, ce signal étant directement corrélé au déplacement de l'impulsion électrique dans les fibres du muscle cardiaque.

L'activité électrique instantanée peut être définie par un vecteur orienté suivant la différence de potentiel présente dans le cœur, et de module proportionnel à celle-ci. Le couple d'électrodes enregistre à chaque instant l'amplitude de la projection de ce vecteur suivant leur axe: ainsi, lorsque le vecteur électrique est orienté de l'électrode « - » à l'électrode « + », on observe sur l'enregistreur une déflexion positive, et lorsque le vecteur est orienté en sens inverse, la déflexion est négative. À l'échelle d'une cellule, le détail des tracés est indiqué sur la Figure ci dessous.

Explication de la figure 6

Considérons une cellule unique ; au repos, elle est chargée uniformément ; le vecteur de dépolarisation est nul (point rouge), le tracé est donc plat (a). Une stimulation extérieure du côté gauche induit une perte de charge de ce côté ; l'impulsion électrique se propage alors de gauche à droite. Le vecteur de dépolarisation associé (flèche rouge) est orienté de l'électrode négative vers l'électrode positive : l'enregistrement présente donc une déflexion positive (b) qui est maximale lorsque la dépolarisation a atteint le milieu de la cellule. La fin de la dépolarisation se traduit par une pente descendante (c), car le vecteur est toujours orienté dans le même sens mais son amplitude diminue. Une fois la cellule dépolarisée, le tracé est plat (d). La re-polarisation de la cellule se traduit par un vecteur électrique orienté dans le sens opposé au précédent ; le tracé présente donc, dans un premier temps, une déflexion négative (e) pour ensuite redevenir plat (f) [Hurst, 1990].

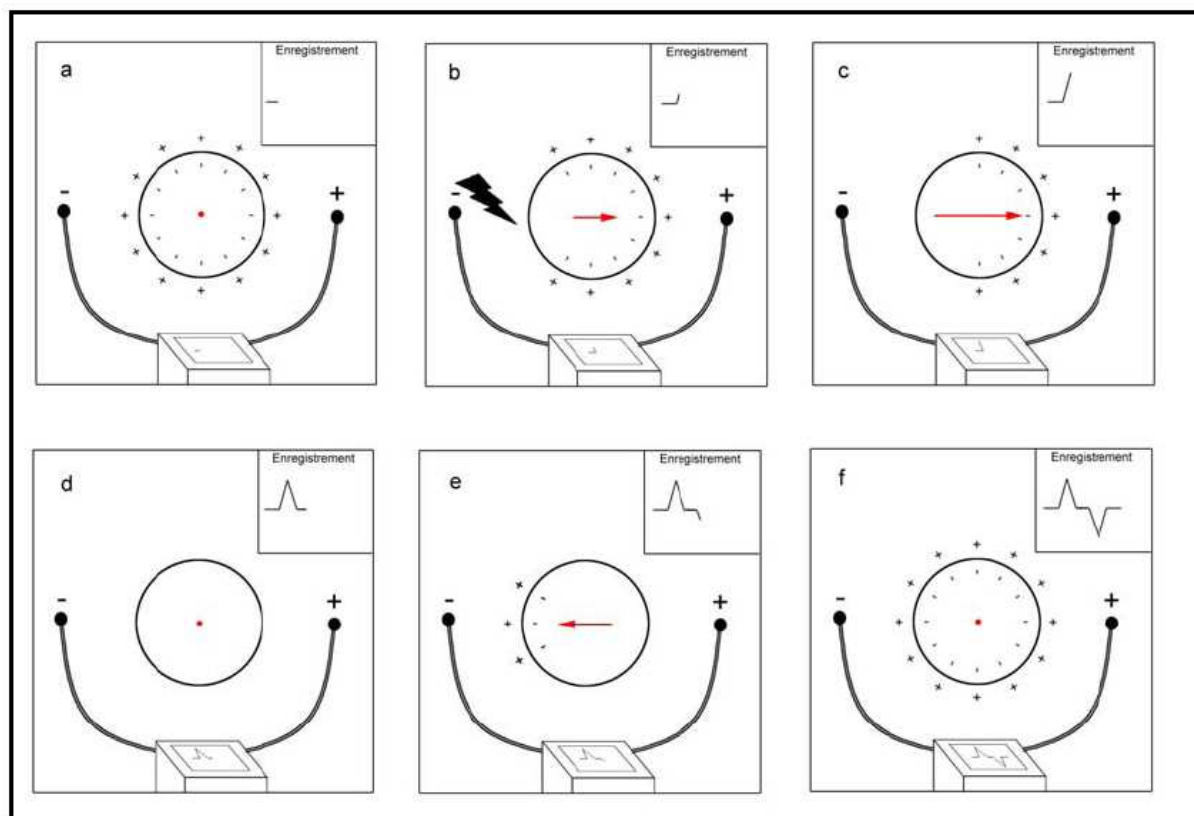


Figure 17 : principe de fonctionnement de l'ECG [Hurst, 1990]

3. Caractéristiques d'un battement cardiaque

L'activité du coeur est enregistrée par des signaux électriques à la surface du corps à des points spécifiques, qui constitue l'électrocardiogramme (ECG). L'ECG est le signal biomédical le plus étudié pour caractériser les anomalies cardiaques [Alg 00]. L'ECG peut être divisé approximativement en trois composantes principales **P**, **QRS** et **T** (figure 7). Le rythme cardiaque normal est composé par des ondes liées en général à des actions mécaniques du coeur.

Les différentes ondes et les intervalles les séparant ont permis de dresser le standard ECG. Toute interprétation de l'électrocardiogramme se réfère aux caractéristiques de forme et de largeur d'onde du signal ECG normal présentées sur la figure 7.

- L'onde **P** représente la dépolarisation auriculaire depuis le noeud sinusal vers le noeud atrio-ventriculaire. C'est l'onde qui précède le complexe **QRS**.
- Le complexe **QRS** correspond à la dépolarisation ventriculaire précédant l'effet mécanique de contraction, l'onde **R** est de grande amplitude car la masse des ventricules est supérieure à celle des oreillettes. Par définition, l'onde **Q** est la première onde négative, l'onde **R** la première onde positive du complexe, et l'onde **S** la première onde négative après l'onde **R**. Toute onde supplémentaire, positive ou négative, sera appelée **R'**, **S'**, **R''**, etc. La figure 8 donne les différents aspects que peut prendre le **QRS**.
- L'onde **T** représente la repolarisation des ventricules. Cette onde succède au complexe **QRS** après retour à la ligne isoélectrique, (sauf pathologie particulière).
- Intervalle **RR** : fréquence des battements cardiaques ; il correspond au délai entre deux dépolarisations des ventricules. C'est cet intervalle qui permet de calculer la fréquence cardiaque.
- Intervalle **PP** : période de polarisation des oreillettes et c'est le délai entre deux dépolarisations des oreillettes.
- Segment **PR** : pause du noeud AV (auriculo-ventriculaire). Le segment **PR** correspond au délai entre la fin de la dépolarisation des oreillettes et le début de celle des ventricules. C'est le temps pendant lequel l'onde de dépolarisation est bloquée au niveau du noeud AV.
- Intervalle **PR** : durée de conduction auriculo-ventriculaire ; cet intervalle correspond au délai entre le début de la dépolarisation des oreillettes et celle des ventricules. C'est le temps de propagation de l'onde de dépolarisation jusqu'aux cellules myocardiques ventriculaires.
- Intervalle **QT** : durée de systole ventriculaire, il représente le temps de systole ventriculaire, qui va du début de l'excitation des ventricules jusqu'à la fin de leur relaxation.
- Segment **ST** : durée de stimulation complète des ventricules, ce segment correspond à la phase pendant laquelle les cellules ventriculaires sont toutes dépolarisées, le segment est alors isoélectrique.
- Enfin, l'onde **U** non constante, traduit la repolarisation du réseau (des fibres) de Purkinje [Ham 04]. C'est une onde positive qui suit l'onde **T**, visible essentiellement en précordiale, dont la signification est discutée (repolarisation prolongée des cellules du réseau de Purkinje ou à un facteur mécanique correspondant à la relaxation du myocarde).

En général, les anomalies cardiaques laissent leurs traces sur un ou plusieurs paramètres liés aux ondes de l'ECG.

L'enregistrement de l'évolution temporelle du champ électrique résultant de l'activité électrique du coeur est appelé l'électrocardiogramme de surface (ECG). L'ECG est effectué au moyen d'électrodes cutanées, sur un support papier standard. Les plans frontal et horizontal représentent respectivement les dérivations des membres et les dérivations précordiales.

En largeur 1 mm équivalent à une durée de 40 ms et en hauteur 1 mm correspond à 0,1 mV mais d'habitude on exprime les voltages en mm [Ham 04].

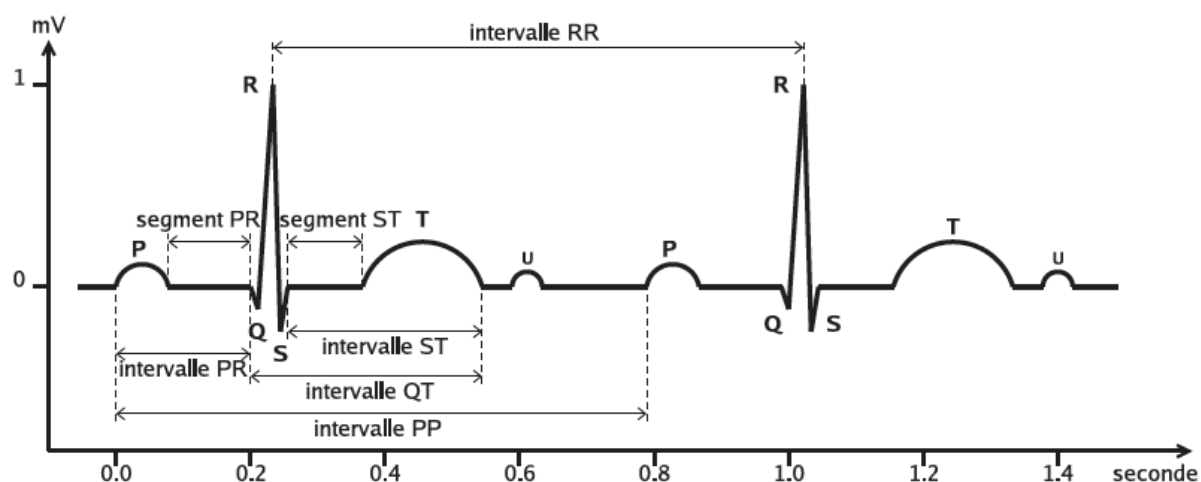


Figure 18 : ECG normal

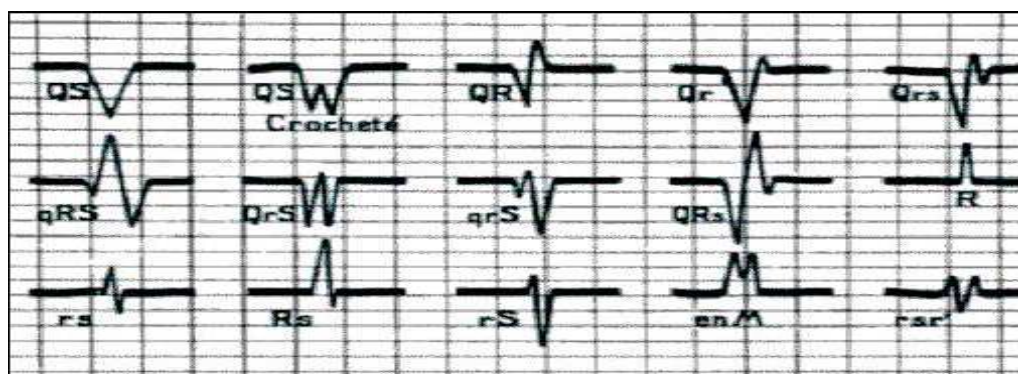


Figure 19 : Schémas des différents aspects du complexe QRS

Les fréquences relatives à chaque onde présentent des variations selon le rythme cardiaque. Le changement du rythme de battement est appelé Arythmie. La bande fréquentielle des signaux ECGs est de 60 Hz environ pour un sujet normal et peut aller jusqu'à 130 Hz pour un patient anormal [Ben MESSAOUD, 2008].

Le tableau suivant donne les actions des membranes du coeur dans le cas normal et les ondes associées ainsi que leurs caractéristiques temporelles et fréquentielles [Ram 03] [Ham 04].

<i>Actions mécaniques</i>	<i>Onde associée</i>	<i>Durée (ms)</i>	<i>Amplitude (mV)</i>	<i>Fréquence de l'onde (Hz)</i>
<i>Contraction des oreillettes (ou dépolarisation auriculaire)</i>	<i>Onde P</i>	<i>80-120</i>	<i>0,25</i>	<i>10</i>
<i>contraction (ou dépolarisation) des ventricules</i>	<i>Complexe QRS</i>	<i>85-120</i>	<i>Q<0 R>0; (0,5-2) S<0</i>	<i>20-50</i>
<i>Retour à l'état de repos (ou repolarisation) des ventricules</i>	<i>Onde T</i>	<i>200</i>	<i>0,2</i>	<i>5</i>
<i>Repolarisation des oreillettes</i>	<i>Onde cachée par le complexe QRS</i>			

Tableau 2 : Caractéristiques mécaniques et électriques d'une arythmie normale

4. Rôle de l'électrocardiogramme

L'ECG permet la détection et l'analyse de la gravité de nombreuses maladies cardiaques. Il permet aussi d'étudier beaucoup plus précisément les ondes auriculaires [Mag 02] et ventriculaires [Jek 00]. Les ondes auriculaires sont les ondes **P**. Les ondes ventriculaires sont les ondes **T**. Si le coeur bat lentement (fréquence cardiaque < 50 battements par minute) c'est la Bradycardie; Le rythme accéléré (fréquence cardiaque > 100 battements par minute) correspond à la Tachycardie. La Tachycardie Ventriculaire est marquée suite à une cicatrice d'une crise cardiaque antérieure. La Fibrillation Auriculaire décrit l'activation totalement chaotique dans les oreillettes. Dans ce cas, la production des fluctuations est irrégulière dans la ligne de base alors que le complexe **QRS** reste normal [Ham 04].

V. Troubles du rythme cardiaque

1. Le rythme cardiaque

L'étude du rythme cardiaque se fait à partir du repérage des ondes **R** ; ce rythme est caractérisé par deux propriétés : la fréquence des ondes **R**, exprimée en nombre de battements par minute (bpm), et leur régularité.

En l'absence de toute pathologie, le rythme est régulier et sa fréquence est comprise entre 60 et 100 bpm la journée et 40 et 80 bpm la nuit. Hors de ces limites, il peut y avoir trouble du rythme qui doit faire l'objet d'une étude approfondie pour définir une éventuelle pathologie (diagnostic à partir du rythme). Cependant, toute irrégularité n'est pas pathologique : en effet, le système nerveux autonome, exerçant un contrôle permanent, peut fortement accélérer le rythme en réponse à un contexte particulier : une période de stress (figure 9) ou d'effort, par exemple. Il est donc essentiel de prendre en considération l'activité du patient avant de poser un diagnostic.



Figure 20 : Le rythme cardiaque est en permanence soumis au contrôle du système nerveux. Ici, au cours d'une épreuve d'examen oral, le rythme cardiaque d'un étudiant atteint 135 bpm, ce qui reste normal étant donné le contexte.

2. Les arythmies cardiaques

Pour dresser un diagnostic à partir d'un enregistrement ECG, deux points essentiels sont à étudier : le rythme cardiaque (fréquence et régularité) et la forme des ondes (distances relatives, amplitudes, durées,...).

L'analyse du rythme ne nécessite que le repérage des ondes **R** : elle fut donc l'un des premiers traitements automatiques du signal cardiaque. Cette analyse est fondée sur l'extraction, à partir du signal, des deux paramètres caractéristiques suivants : la fréquence des battements et leur régularité. Il existe certaines pathologies induisant ces deux anomalies rythmiques.

2.1. Extrasystole ventriculaire

L'extrasystole ventriculaire (ESV) est un battement anormal suffisamment répandu pour être présenté en introduction. Les ESV s'observent sur quasiment tous les enregistrements, principalement en période de récupération après un effort. Bien que leur présence n'indique aucune pathologie particulière, si de façon récurrente, leur nombre par minute est supérieur à 6, elles peuvent être un signe précurseur d'une tachycardie ventriculaire, qui peut constituer une pathologie majeure.

Contrairement aux battements normaux qui ont pour origine la dépolarisation des cellules sinusales, l'ESV naît de la dépolarisation spontanée d'un petit groupe de cellules ventriculaires, appelé alors foyer ectopique ventriculaire. L'impulsion électrique créée n'emprunte pas la voie normale de conduction (faisceau de His), et se propage donc plus lentement dans les ventricules. La contraction ventriculaire ainsi étalée dans le temps perd de son efficacité.

Le tracé d'un battement ESV est caractérisé par deux propriétés : l'onde **R** n'est pas précédée d'une onde **P**, puisqu'il n'y a pas eu d'activité auriculaire préalable, et la durée du complexe est supérieure à la durée d'un complexe **QRS** normal (figure 10).

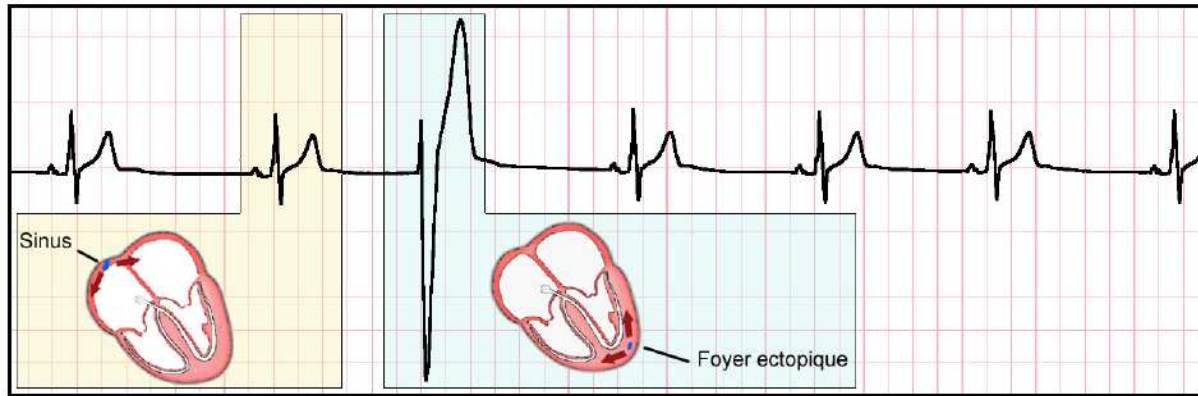


Figure 21 : extrasystole ventriculaire

2.2. Arythmies à base de fréquence

2.2.1. Bradycardie

La bradycardie est caractérisée par une fréquence cardiaque inférieure à 60 bpm ; elle peut être d'origine sinusale, jonctionnelle, ou ventriculaire, selon le site d'initiation de l'impulsion électrique à l'origine des battements considérés.

a. Bradycardie sinusale

En plus de sa basse fréquence, la bradycardie sinusale est caractérisée par la présence systématique d'une onde **P** avant les complexes **QRS** (figure 11) car l'origine de la contraction ventriculaire reste la dépolarisation du sinus et des oreillettes, comme lors de battements normaux. Les causes d'une telle arythmie sont multiples et souvent extérieures au système cardiovasculaire : traitement médicamenteux (bêtabloquant, digoxine, ...), hypothermie, urémie, ...



Figure 22 : Bradycardie sinusale. Le rythme est de l'ordre de 40 bpm, la présence de l'onde P nous assure de l'origine auriculaire des battements.

b. Bradycardie jonctionnelle

En cas de dysfonctionnement sinusal avec ou sans dépolarisation auriculaire, le noeud AV peut assurer le rôle de pacemaker de secours à une fréquence de l'ordre de 30 à 60 bpm. L'impulsion électrique prend naissance dans le noeud auriculo-ventriculaire (AV) et suit le chemin de conduction habituel : la morphologie des ondes **QRS** est identique à celle du battement normal ; en revanche, l'onde **P** peut être absente ou, si elle est présente, elle peut être désynchronisée de la systole ventriculaire (figure 12).

- Ce rythme, appelé rythme d'échappement jonctionnel, dure jusqu'à ce qu'il soit inhibé par un rythme plus rapide.

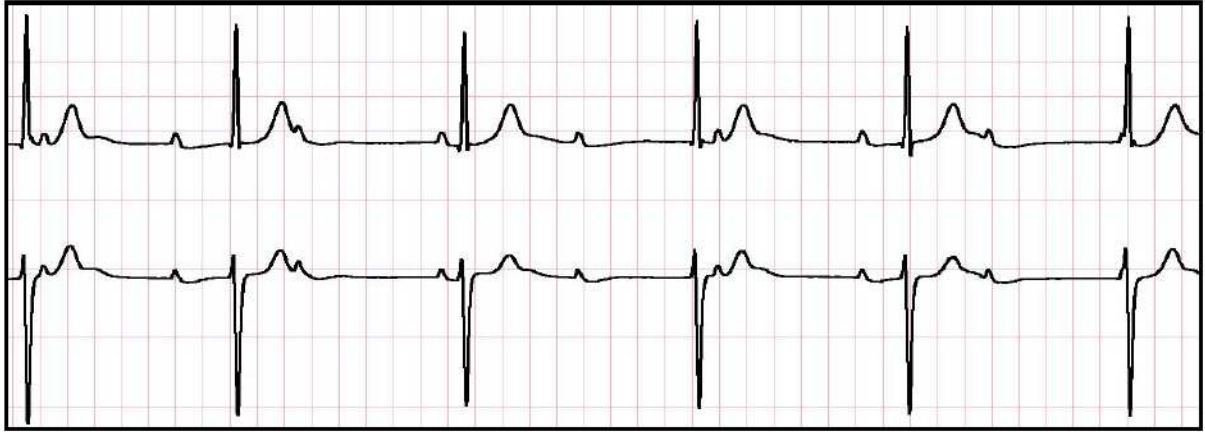


Figure 23 : Rythme d'échappement jonctionnel lent à 35 bpm.

- Lorsque le problème de conduction intervient en aval du noeud AV, c'est un groupe de cellules du muscle ventriculaire qui est susceptible de devenir pacemaker ; le rythme devient alors une succession d'ESV à une fréquence très lente, entre 15 et 40 bpm, appelé rythme d'échappement ventriculaire. Ce type de pathologie constitue une indication typique de la pose d'un stimulateur cardiaque pour éviter une syncope possible.

2.2.2. Tachycardie

À l'inverse de la bradycardie, la tachycardie est caractérisée par le fait que la fréquence est supérieure à 100 bpm ; elle peut être d'origine sinusale, auriculaire ou ventriculaire.

a. Tachycardie sinusale

La tachycardie sinusale correspond à un rythme sinusal dont la fréquence est comprise entre

100 et 180 bpm ; une onde **P** est observée avant chaque complexe **QRS**, de forme identique à celle observée lors du rythme normal. Il est déjà mentionné sur la figure 6 qu'une «tachycardie » modérée et apparaissant dans des conditions de stress ou d'effort est physiologique. Une tachycardie sinusale est considérée comme pathologique si elle est de longue durée et indépendante du contexte (figure 13).

Les causes de cette pathologie sont habituellement extracardiaques, et incluent tous les facteurs de stimulations du système nerveux (surmenage,...) et l'influence de différentes substances comme l'adrénaline ou la caféine.

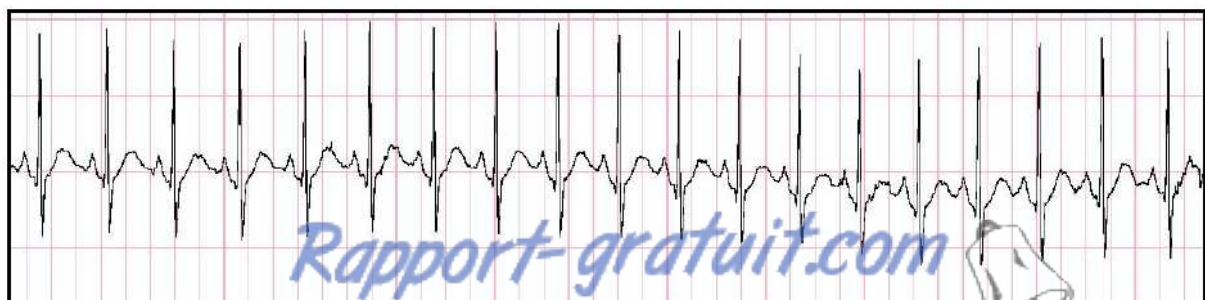


Figure 24 : Tachycardie sinusale

b. Tachycardie auriculaire et nodale AV

La tachycardie auriculaire peut avoir pour origine un foyer ectopique, une boucle de stimulation (flutter) ou une voie qui court-circuite la voie AV, dite voie accessoire, avec réentrée par le noeud AV. Dans le cas d'un foyer ectopique, il s'agit d'un groupe de cellules situées dans les oreillettes, qui se dépolarisent spontanément et plus rapidement que le sinus, prenant ainsi sa place. La dépolarisation des oreillettes n'étant pas d'origine sinusale, la propagation de l'influx nerveux diffère de celle qui a pour origine le sinus, et l'on observe une onde **P** de forme inhabituelle.

La fréquence typique de décharge de ce type de foyer est comprise entre 120 et 200 bpm. La décharge régulière d'un foyer ectopique localisé dans le noeud AV peut aussi être la cause d'une tachycardie, dite tachycardie nodale AV ; dans ce cas, la fréquence des battements peut atteindre 250 bpm. Contrairement à la tachycardie auriculaire, aucune onde **P** ne précède les complexes **QRS**, car il n'y a pas d'activité auriculaire avant le battement (il peut y avoir une activité auriculaire rétrograde dont la trace sur l'ECG est noyée dans le complexes **QRS**).

Le principal risque de ce type de pathologie est le manque d'efficacité des ventricules qui, contraints de se contracter très fréquemment, n'ont pas le temps de se remplir correctement de sang : l'alimentation du corps en oxygène peut en être altérée.

c. Tachycardie ventriculaire (TV)

La tachycardie ventriculaire a pour origine un ou plusieurs foyer(s) ectopique(s) ventriculaire(s) (qui se dépolarisent à tour de rôle). Les battements ont donc la forme d'extrasystoles ventriculaires très rapprochées (figure 14). Ce type de rythme est dangereux à cause de son évolution en fibrillation ventriculaire qui conduit au décès du patient si elle n'est pas traitée à l'aide d'un défibrillateur dans les quelques minutes qui suivent son apparition.

Les bradycardies et tachycardies engendrent un rythme régulier pendant leur manifestation. Cependant, certains de ces événements peuvent, d'une part, se manifester de manière sporadique par phases, et produire ainsi des rythmes différents d'assez longues durées, ou d'autre part, créer de véritables troubles de la régularité.



Figure 25 : Tachycardie ventriculaire

2.3. Les arythmies à base de régularité

L'absence de régularité des battements cardiaques est une caractéristique du rythme importante pour le diagnostic ; elle est souvent associée à un trouble de la production ou de la conduction de l'impulsion électrique (foyers ectopiques, blocs, boucles, ...). Les arythmies permanentes ou sporadiques nécessitent un suivi médical et sont des indications typiques à la pose régulière d'un enregistreur ECG (ou Holter).

2.3.1. Foyer ectopique auriculaire ou nodal AV

Les irrégularités de rythme peuvent traduire la présence d'un ou plusieurs foyers ectopiques auriculaires (figure 15). La fréquence d'expression de ces foyers, leur alternance entre eux et avec le sinus, et la transmission aux ventricules, peuvent entraîner diverses conséquences rythmiques : de l'extrasystole auriculaire isolée (ESSV) avec un repos compensatoire, qui introduit une distorsion rythmique locale, jusqu'à la fibrillation auriculaire évoquée plus loin, en passant par une tachycardie régulière ou une tachyarythmie (bigéminisme).

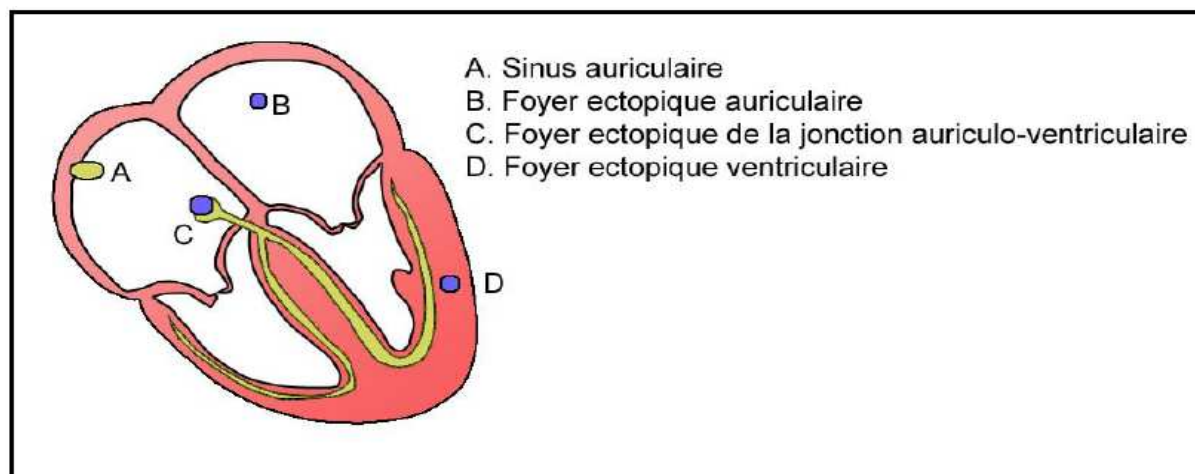


Figure 26 : Emplacement des foyers ectopiques.

2.3.2. Foyer ectopique ventriculaire

Lorsque le foyer ectopique se trouve dans les ventricules, on obtient un battement ESV. Les ESV étant fréquemment suivies d'un repos compensatoire, sorte de temps de récupération, elles introduisent en général une distorsion locale du rythme, qui peut réapparaître plus ou moins fréquemment ou régulièrement selon la fréquence d'expression du ou des foyers ectopiques, et leur caractère aléatoire ou non (figure 16).

Bigéminismes et Trigéminismes

Les bigéminismes et trigéminismes sont des rythmes à deux commandes. La commande de base (généralement sinusale) est interrompue par des battements d'origine ectopique. Lorsque l'on se trouve en présence d'un bigéminisme les **QRS** qui appartiennent au rythme de base sont suivis d'un **QRS** d'origine ectopique avec une succession de 1/1. On parle de trigéminisme lorsqu'on est en présence d'une succession 2/1. Les figures 16 et 17

présentent respectivement un exemple de rythme de bigéminisme et de trigéminisme dans le cas d'un rythme sinusal avec un foyer ectopique ventriculaire.

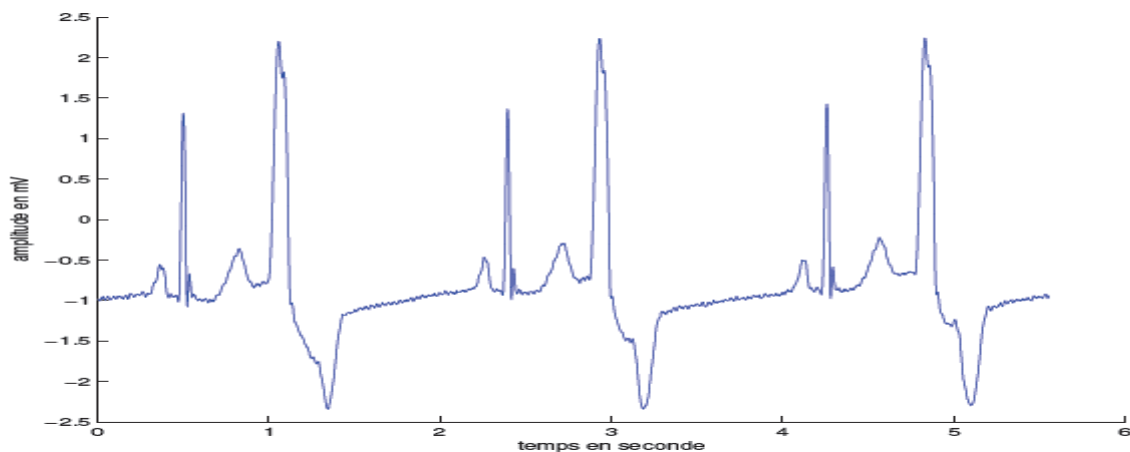


Figure 27: Exemple de rythme de Bigéminisme

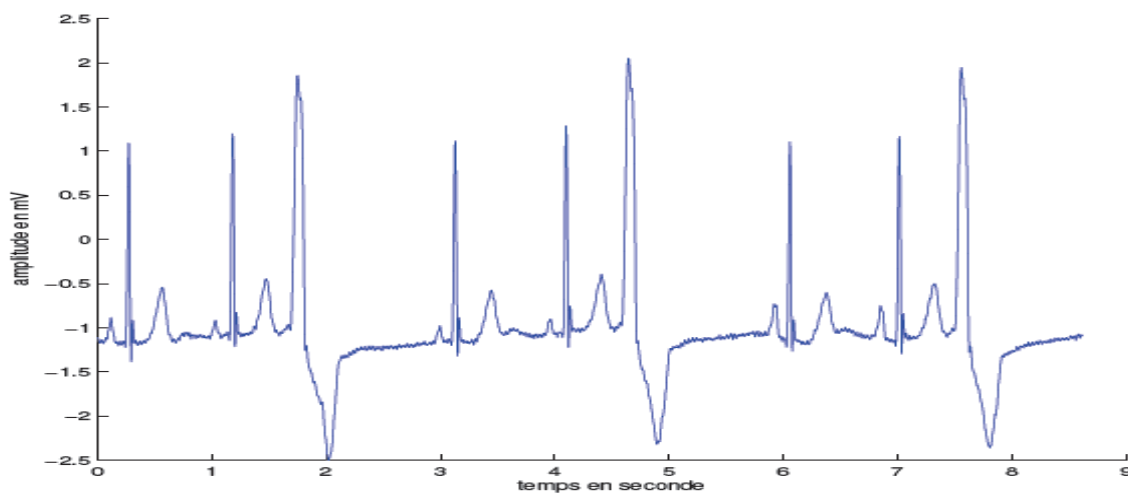


Figure 28 : Exemple de rythme de Trigéminisme

2.3.3. Fibrillations et Flutter

a. Flutter auriculaire

Le flutter est parmi les causes de troubles du rythme, dans ce cas, la fréquence de l'onde P peut atteindre 300 bpm, voire davantage. À cette fréquence, le noeud auriculo-ventriculaire ne parvient pas à conduire toutes les impulsions électriques vers les ventricules, et l'on observe généralement des blocs 2:1, au cours desquels seule une contraction auriculaire sur deux est suivie d'une systole ventriculaire ; la fréquence ventriculaire est donc d'environ 150 bpm dans un tel cas.

Plus rarement, des flutters 3:1, 4:1 sont observés. Dans ce type de pathologie, il existe un risque majeur pour la santé du patient, en raison de la possibilité de génération d'embolies, dus aux turbulences dans le flux sanguin au niveau des oreillettes.

b. Fibrillation auriculaire (FA)

La fibrillation auriculaire est plus fréquente que le flutter, elle touche 5 à 10% des personnes de plus de 65 ans. Dans cette pathologie, l'activité auriculaire n'est plus composée de battements ; soumise à plusieurs foyers ectopiques et à des boucles de conduction locales, le fonctionnement du myocarde auriculaire est totalement désorganisé. Ces mouvements anarchiques peuvent néanmoins transmettre quelques impulsions au noeud auriculo-ventriculaire, qui les transmet à son tour aux ventricules et entraîne leurs contractions. Comme cette transmission revêt un caractère aléatoire, le rythme des complexes QRS est complètement irrégulier. L'absence des systoles auriculaires n'est pas grave en soi, mais elle est cependant responsable d'une baisse significative de l'efficacité cardiaque à deux niveaux :

- D'une part, parce que le coeur ne bénéficie pas de la systole auriculaire qui assure une partie du remplissage sanguin des ventricules,
- Et d'autre part, parce que le rythme moyen est généralement supérieur à 100 bpm et peut atteindre 200 bpm.



Figure 29 : Fibrillation auriculaire

Le risque majeur lié à cette pathologie est celui du flutter, c'est-à-dire de l'envoi possible, dans la circulation, d'embolies formées au niveau des oreillettes. Ce risque est relativement faible lorsqu'une FA est permanente, tandis qu'il est augmenté lors d'épisodes de FA sur fond sinusal (FA paroxystique), surtout lors du passage d'un rythme à l'autre.

c. Fibrillation ventriculaire (FV)

Elle représente l'équivalent physiologique de la fibrillation auriculaire, mais transposée aux ventricules : les ventricules se déchargent alors de manière totalement désynchronisée, et il n'y a plus de systole cardiaque. La fibrillation ventriculaire constitue une arythmie particulièrement grave, car elle est une menace de mort imminente : en effet le coeur n'assure plus du tout son travail de pompe, le sang ne circule plus, ce qui conduit à une asphyxie de tous les tissus du corps, dont le myocarde lui-même. Sans une intervention (défibrillation) immédiate, susceptible de re-synchroniser la dépolarisation des cellules du myocarde et faire ainsi « repartir » le mouvement cardiaque, la mort s'ensuit. Les personnes qui présentent de tels risques peuvent aujourd'hui bénéficier de l'implantation d'un défibrillateur : placé au niveau du thorax (pacemaker), il est muni d'une sonde qui peut détecter l'anomalie rythmique et conduire l'appareil à délivrer une forte décharge électrique.

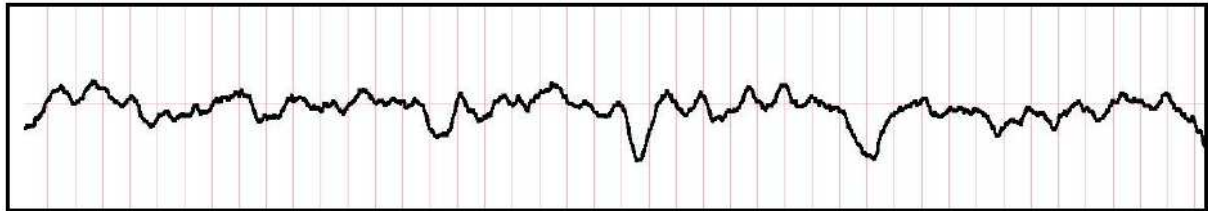


Figure 30 : Fibrillation ventriculaire

2.3.4. Blocs cardiaques

Le bloc est un exemple de problèmes de conduction. C'est un défaut localisé de propagation de l'impulsion électrique dans le tissu cardiaque. Lorsqu'il est complet, c'est-à-dire que l'absence de conduction est totale, des pacemakers de réserve comme un foyer ectopique auriculaire, ou le noeud AV, peuvent prendre le relais et entraîner des bradycardies ou tachycardies. A l'inverse, lorsqu'il apparaît de manière sporadique, suivant le cas, il peut se manifester par des troubles du rythme. Les quatre types de blocs cardiaques seront détaillés par la suite :

a. Bloc Sino-Auriculaire (Bloc SA)

Le noeud sinusal peut ne pas transmettre de stimulus aux cellules des oreillettes. La conséquence est qu'au moins un cycle complet n'est pas effectué. Après la pause, due au bloc, le cycle reprend normalement si aucun autre foyer ectopique n'a déclenché de contraction.

b. Bloc Auriculo-Ventriculaire (BAV)

On appelle BAV l'altération de la conduction du stimulus de dépolarisation entre les oreillettes et les ventricules. Trois degrés de sévérité sont distingués :

- Les BAV de premier degré provoquent l'allongement du segment **PR** de façon égale à chaque cycle.
- Les BAV de deuxième degré traduisent l'absence momentanée d'onde **QRS** après une onde **P** normale. Lorsque les segments **PR** précédents sont normaux, et c'est le Mobitz de type II (figure 20). Lorsque les segments précédents vont en s'augmentant, on parle de phénomènes de Wenckebach (ou Mobitz de type I).

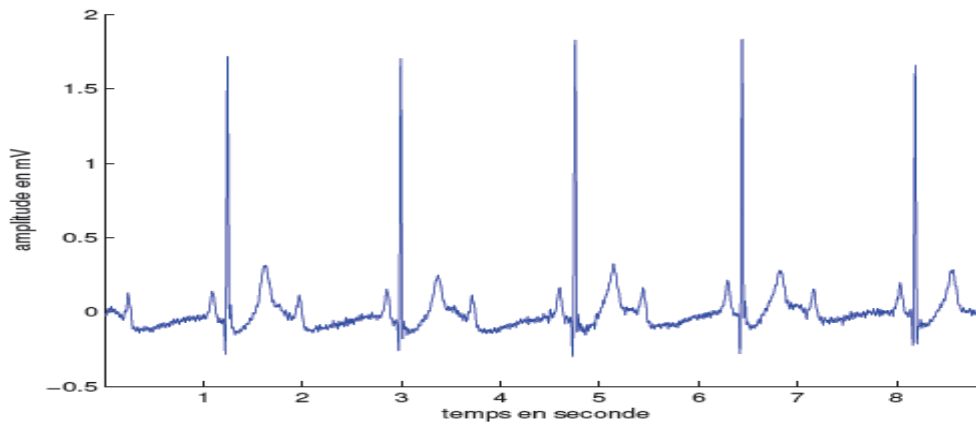


Figure 31 : Exemple de Mobitz de type II

- Les BAV de troisième degré sont dit complets, c'est-à-dire qu'aucune dépolarisation auriculaire ne parvient aux ventricules. Un foyer ectopique ventriculaire ou jonctionnel joue alors le rôle de pacemaker. Le foyer est identifiable par la forme et la fréquence des battements. Les activités auriculaires et ventriculaires sont complètement dissociées.

c. Bloc de branche

Le bloc de branche est dû au blocage de la dépolarisation dans une des branches du faisceau de His. Un bloc dans l'une ou l'autre branche provoque un retard dans la dépolarisation du ventricule auquel elle appartient. Et par la suite la dépolarisation des ventricules est désynchronisée et le complexe **QRS** est élargi (figure 21).

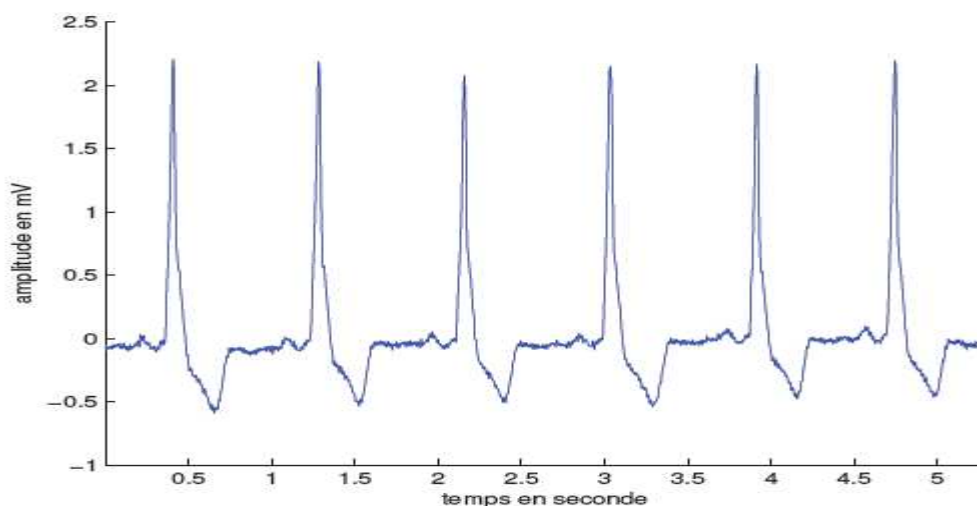


Figure 32 : Exemple de bloc de branche gauche.

d. Syndrome de Wolff-Parkinson-White

Ce syndrome a été décrit en 1930 par l'équipe Wolff-Parkinson-White. Il rentre dans le cadre des phénomènes de "conduction accélérée". Ce syndrome rare touche surtout les hommes (70 %), plutôt jeunes, et son pronostic est en général bénin. Il existe dans ce type de syndrome une voie de conduction anormale entre les oreillettes et les ventricules, dite voie

accessoire. Cette voie court-circuite le nœud auriculo-ventriculaire, entraînant une excitation précoce des ventricules : c'est le phénomène de pré-excitation ventriculaire. On voit sur le tracé ECG un espace **PR** court (< 0.10 seconde) et un complexe **QRS** élargi (>0.12 seconde) [CHIKH 2005].

VI. Conclusion

Ce chapitre a permis de définir les différents concepts de base dans le domaine de cardiologie, tels que l'anatomie du cœur et les composants du système cardiovasculaire, ainsi que les caractéristiques principales du signal ECG.

Aussi, les troubles du système cardiaque ont été brièvement décrites, citant par exemple les troubles de conduction et les arythmies.

Dans notre cadre de recherche de mémoire de Magistère, nous avons ciblé les battements cardiaques normaux "N", et les battements ventriculaires prématurés "BVP", qui sont disponibles dans la base de données cardiologiques de MIT BIH [<http://www.physionet.org/physiobank/database/mitdb>].

Chapitre 4 : Résultats et interprétations

I. Introduction

L'objectif majeur de notre étude est la conception d'un ensemble de réseaux de neurones (RNs) multi couches (une couche d'entrée, une couche cachée et une couche de sortie), afin de pouvoir classer les battements cardiaques, et par la suite évaluer les performances obtenues des différents RNs, selon l'algorithme d'apprentissage adopté et la base d'exemples choisie.

Ce modèle de RN se caractérise par :

- Un apprentissage discriminant : l'apprentissage diminue l'erreur de la classification en respectant la structure des exemples dans la base d'apprentissage, contrairement aux autres méthodes statistiques.
- La puissance d'approximation : les RNs multicouches sont capables d'approximer toute fonction de transfert avec un nombre de neurones cachés (NC) adéquat.

Les RNs conçues sont destinés à reconnaître les pathologies cardiaques dites extrasystoles ventriculaire ESV (battement ventriculaire prématuré, BVP), disponibles dans la base de données MIT BIH [<http://www.physionet.org/physiobank/database/mitdb>].

II. Les bases de données médicales

Tout traitement automatisé ne soit valide que s'il est appliqué sur une collection de données adéquate quelque soit son type (physique, météologique, médicale ...).

Dans notre cas, nous nous intéressons aux bases de données médicales, et plus précisément les bases de données cardiologiques (ensemble de signaux ECG). Ces dernières doivent être les plus représentatives que possible, et couvrent une large variété de pathologies existantes.

Différentes bases de données cardiologiques existent dans le monde médical (tableau 1):

Bases de données	Description
AHA DB	The American Heart Association Database for Evaluation of Ventricular Arrhythmia Detectors. Elle comporte 80 enregistrements de 35 minutes chacun.

CU DB	The Creighton University Ventricular Sustained Arrhythmia Database. Elle comporte 35 enregistrements de 8 minutes chacun.
ESC DB	The European Society of Cardiology ST-T Database. Elle comporte 90 enregistrements de 120 minutes chacun.
MIT DB	The Massachusetts Institute of Technology-Beth Israel Hospital Arrhythmia Database. Elle contient 48 enregistrements de 30 minutes chacun.
NST DB	The Noise Stress Test Database. Elle contient 12 enregistrements de 30 minutes chacun.

Tableau 3 : Quelques bases de données cardiologiques.

6. La base de données MIT BIH

Depuis 1975, les laboratoires de l'hôpital de Beth Israël à Boston et MIT ont réalisé une base de données MIT-BIH, qui a été commencée à être distribuée en 1980. Cette base de données contient 48 enregistrements extraits d'une demi-heure des enregistrements ambulatoires à deux voies d'ECG, obtenus à partir de 47 sujets étudiés par le laboratoire d'arythmie de BIH entre 1975 et 1979. Vingt-trois enregistrements ont été choisis au hasard d'un ensemble de 4000 enregistrements ambulatoires de 24 heures d'ECG rassemblées d'une population mélangée des patients hospitalisés (60%) et des patients non hospitalisés (40%) à l'hôpital de Beth Israël à Boston (les séries '100'), les 25 enregistrements restants ont été choisis parmi les mêmes enregistrements mais qui en considération des arythmies rarement observés qui ont une signification clinique (les séries '200').

Les enregistrements ont été échantillonnés à une fréquence $f_e=360$ Hz avec une résolution de 11 bits sur une gamme de 10mV. Deux cardiologues ou plus ont indépendamment annoté chaque enregistrement, environ 110.000 annotations ont été inclus avec la base de données [<http://www.physionet.org/physiobank/database/mitdb>].

7. Base d'exemples utilisée

Pour réaliser nos expériences d'apprentissage et de test, nous avons sélectionné quelques enregistrements de la base de données MIT BIH, ces enregistrements comprennent suffisamment les arythmies ciblées à classer (BVP).

Le tableau suivant présente les enregistrements, choisis pour réaliser notre application.

Enregistrements	Battements N	Battements V	Nombre total de battements
100	2260	1	2261
101	1855	1	1856
103	2071	-	2071
105	2525	84	2609

106	1500	517	2017
108	1806	83	1889
109	2484	29	2513
111	2104	7	2111
112	2521	-	2521
113	1786	-	1786
114	1814	56	1870
115	1939	-	1939
116	2273	109	2382
117	1526	-	1526
118	2256	13	2269
119	1535	442	1977
121	1851	1	1852
122	2462	-	2462
124	1577	34	1611
200	1815	773	2588
201	1665	260	1925
202	2091	33	2124
203	-	349	349
205	-	68	68
207	-	289	289
208	-	1216	1216
209	2982	9	2991
210	2449	180	2629
212	2731	-	2731
213	2948	280	3228
214	2013	235	2248
215	3194	153	3347
219	2083	57	2140
220	2036	-	2036
221	2022	393	2415
222	2478	32	2510
223	2173	416	2589
228	1706	363	2069
230	2238	1	2239
231	1555	2	1557
232	-	6	6
233	-	811	811
234	-	3	3

Tableau 4 : Caractéristiques des enregistrements de la base d'exemples utilisée.

III. Extraction des descripteurs

Chaque enregistrement se compose d'un ensemble de paramètres cliniques. La précision des résultats de reconnaissance et de classification des anomalies se repose sur le choix adéquat des paramètres relatifs à chaque classe. Pour la sélection des ces paramètres, une consultation des experts (en général un médecin) s'avère nécessaire. Une redondance d'information peut apparaître si nous choisissons un nombre élevé de paramètres. De plus ceci nécessite l'utilisation d'un réseau de neurones de taille importante [Hu 94]. Par contre, un nombre réduit de paramètres ne peut pas discerner les classes. Différentes caractéristiques sont extraites à partir de l'ECG pour la classification des arythmies ventriculaires utilisant le complexe QRS et le segment ST, le rythme cardiaque, les caractéristiques spectrales, coefficients d'AR, mesures de complexité et mesures de non linéarité.

Dans [Zon 98], 12 paramètres sont choisis incluant les aires des ondes. Dans [Chr 05], en plus des paramètres classiques (aires, amplitudes et durées), les informations morphologiques sont déployées, le nombre des paramètres atteint 26 [Ben Messaoud 2008].

Plusieurs algorithmes sont utilisés pour la mesure de différents paramètres des ECGs. Le plus répandu est la mesure de l'apparition de l'onde R du complexe QRS [Pan 85].

Dans ce qui suit, nous allons définir les six (6) descripteurs utilisés :

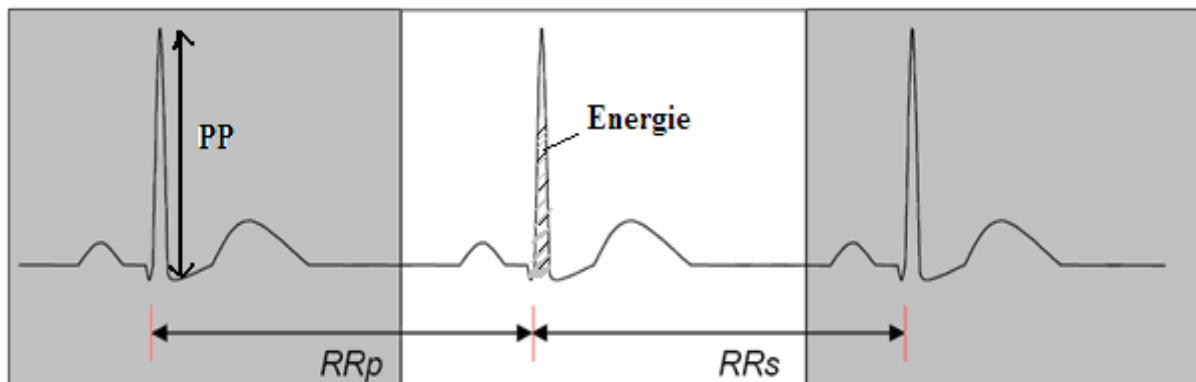


Figure 33 : Intervalles PP, RR_p et RR_s.

1. Intervalle RR_p

L'intervalle **RR_p** représente la distance qui sépare le pic **R** du battement présent et le pic **R** du battement précédent.

2. Intervalle RR_s

L'intervalle **RR_s** représente la distance qui sépare le pic **R** du battement présent et le pic **R** du battement suivant.

3. Rapport des intervalles RR_s/RR_p

Le rapport entre **RR_s** et **RR_p** est au voisinage de 1 si le battement est normal, et il dépasse la valeur de 1 si c'est le cas d'un battement extrasystole ventriculaire (ESV).

4. *Largeur du complexe QRS*

Le complexe **QRS** est la forme d'onde la plus perceptible du signal ECG. Vu son amplitude élevée, sa détection est plus facile que les autres ondes. Ainsi, elle est généralement employée comme référence dans le cycle cardiaque. Dans un battement ventriculaire prématuré (BVP), le complexe **QRS** devient large.

5. *Distance PP (figure 1)*

Elle représente la distance entre le point minimum et le point maximum du complexe **QRS**, sachant que cette distance varie énormément avec la déformation du complexe **QRS**, elle est donc un paramètre informationnel pertinent pour chaque cycle cardiaque.

6. *Energie du complexe QRS (figure 1)*

C'est la moyenne des carrées des échantillons d'un complexe **QRS**, l'aire sous le pic **R**.

IV. Apprentissage des réseaux de neurones

1. *Phase d'apprentissage*

L'apprentissage consiste à calculer les pondérations optimales des différentes connexions du réseau de neurones en utilisant un échantillon de données. A un même type du réseau de neurones, plusieurs scénarios d'apprentissage peuvent être envisagés. La vitesse de convergence (temps d'apprentissage) et l'erreur d'apprentissage sont les deux critères de choix les plus utilisés.

Dans un réseau de neurones multicouche, l'apprentissage se déroule comme suit :

- La couche d'entrée reçoit les données sources utilisées pour l'analyse. Dans le contexte d'aide au diagnostic médical, cette couche recevra les symptômes. Sa taille est donc déterminée par les descripteurs choisis au préalable.
- La couche cachée n'a qu'une utilité intrinsèque pour le réseau de neurones et n'a aucun contact avec l'extérieur.
- La couche de sortie fournit le résultat. Dans notre travail, le battement du patient, est-il une arythmie (BVP) ou un cas normal. Sa taille est définie par le nombre de classes fixé au début.
- L'apprentissage permet une comparaison entre le résultat obtenu en fonction des entrées fournies, et la sortie désirée. Ainsi, le réseau continue à se modifier jusqu'à obtention des résultats satisfaisants.

2. Phase de test (validation)

La phase de test s'effectue après entraînement du réseau de neurones (après apprentissage). Lors de test, la base de données utilisée doit être différente de celle utilisée en phase d'apprentissage.

Une fois le test terminé, quatre situations sont obtenues. En effet, le résultat de l'affectation peut être positif ou négatif, et la pathologie est présente (vrai) ou absente (faux). Deux interprétations des résultats obtenus sont correctes (vrai positif et vrai négatif) ; et les deux autres sont fausses (faux positif et faux négatif). Ces quatre combinaisons peuvent être représentées sous la forme d'une matrice (tableau 3).

Résultats	Vérité	
	Maladie présente	Maladie absente
Classement positif	vrai positif (VP)	faux positif (FP)
Classement négatif	vrai négatif (VN)	faux négatif (FN)

Tableau 5 : Matrice des résultats de test

Les quatre éléments de la matrice ci-dessus permettent de calculer des indicateurs de la qualité des réseaux de neurones utilisés.

Un bon test de diagnostic minimise le nombre de FN et de FP.

- FN (Faux Négatif): désigne un test négatif dans un individu atteint d'une maladie.
- FP (Faux Positif) : désigne un test positif dans un individu sain.
- VN (Vrai Négatif) : désigne un test négatif dans un individu sain.
- VP (Vrai Positif) : désigne un test positif dans un individu atteint d'une maladie.
- Se (Sensibilité) : elle donne la probabilité de détection est définie par [Gao 05] :

$$Se(\%) = 100 \cdot \frac{VP}{VP + FN}$$

- Sp (Spécificité) : c'est le pourcentage d'avoir une interprétation négative quand il n'y a pas de pathologie.

$$Sp(\%) = 100 \cdot \frac{VN}{FP + VN}$$

V. Conception des réseaux de neurones

Nous avons choisi d'utiliser des réseaux de neurones multicouches, avec différentes architectures, fonctions d'activation et algorithmes d'apprentissage.

1. Architectures

- Couche d'entrée : elle comporte six (6) neurones correspondant aux six descripteurs définis précédemment {**RRp**, **RRs**, **RRs/RRp**, **QRS**, **PP**, **Energie**}. Pour tous les réseaux de neurones conçus, le vecteur d'entrée est invariable.
- Couche cachée : le choix de la taille de la couche cachée est souvent réalisé d'une manière empirique, car il n'existe aucune formule d'optimisation. Dans le cadre de notre application, nous avons fait varier le nombre de neurones cachés entre **1** et **15**, ceux retenus selon l'erreur quadratique et le temps de réponse sont : **3, 5, 8, 10** et **15**.
- Couche de sortie : la sortie est représentée par un seul neurone correspondant aux deux classes utilisées "**N=0**" pour le cas normal et "**V=1**" pour le cas ESV. La fonction d'activation de la couche de sortie est une fonction linéaire ($F(x)=x$).

2. Fonctions d'activation

Pour notre cas, nous avons utilisé pour chaque nombre de neurones cachés, les trois fonctions d'activations (sigmoïde, tangente hyperbolique et linéaire) décrites dans le tableau 4.

- Le nombre d'itérations est fixé à **1000** afin de limiter le temps écoulé lors de l'apprentissage, et a permis de réaliser un apprentissage plus ou moins satisfaisant. En plus que s'il était trop grand, il y aurait une situation de sur apprentissage, et par contre s'il était trop petit, un arrêt précoce de l'apprentissage est survenu.
- Le meilleur pas d'apprentissage pour tous les essais est égal à **0.05** sachant qu'un pas d'apprentissage petit provoque assez d'oscillations, et un pas d'apprentissage proche de la valeur de un (1) entraîne des erreurs.
- L'erreur d'apprentissage est fixée à 10^{-3} pour quelques essais, et à 10^{-2} pour d'autres.










Nom de la fonction	Relation d'entrée/sortie	Icône	Nom Matlab
seuil	$a = 0$ si $n < 0$ $a = 1$ si $n \geq 0$		hardlim
seuil symétrique	$a = -1$ si $n < 0$ $a = 1$ si $n \geq 0$		hardlims
linéaire	$a = n$		purelin
linéaire saturée	$a = 0$ si $n < 0$ $a = n$ si $0 \leq n \leq 1$ $a = 1$ si $n > 1$		satlin
linéaire saturée symétrique	$a = -1$ si $n < -1$ $a = n$ si $-1 \leq n \leq 1$ $a = 1$ si $n > 1$		satlins
linéaire positive	$a = 0$ si $n < 0$ $a = n$ si $n \geq 0$		poslin
sigmoïde	$a = \frac{1}{1+\exp^{-n}}$		logsig
tangente hyperbolique	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$		tansig
compétitive	$a = 1$ si n maximum $a = 0$ autrement		compet

Tableau 6 : Exemples de fonction d'activation.

3. Algorithmes d'apprentissage

Pour les réseaux de neurones multicouches, MATLAB dispose de la version basique de l'algorithme de rétropropagation, ainsi qu'une variété d'algorithmes qui varient en coût de calcul et vitesse de convergence, comme l'algorithme de quasi-Newton. Chaque type d'algorithme répond à des besoins spécifiques (classification, approximation...).

La méthode de Levenberg-Marquardt est l'une de ces algorithmes et est devenue un standard pour l'optimisation de l'erreur quadratique grâce à sa robustesse (convergence rapide).

Pour notre expérimentation, nous avons utilisé les deux fameux algorithmes :

- Algorithme de **descente du gradient**.
- Algorithme de **Levenberg-Marquardt**.

VI. Différents scénarios d'expérimentation

1. Phase d'apprentissage

Pour ce faire, nous avons utilisé trois (3) bases de données,

- La première est un mélange entre les battements normaux (N) et les battements ventriculaires prématurés (BVP).
- La deuxième comporte seulement des cas normaux (N).
- Et la dernière, ne contient que des cas pathologiques (V).

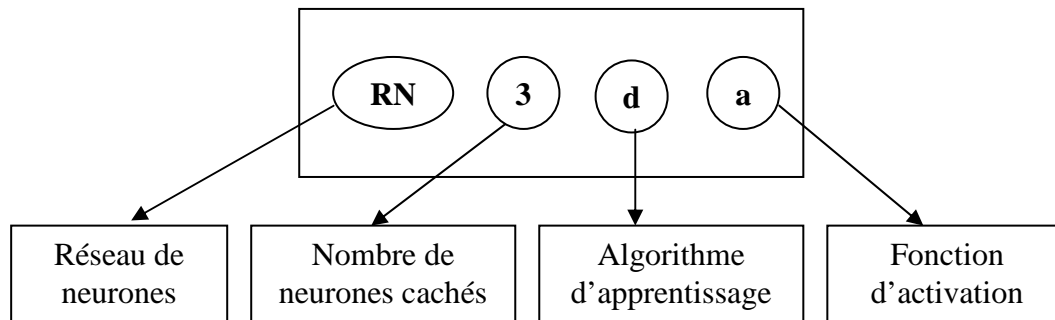
1.1. Apprentissage avec la base mixte

Les résultats obtenus en termes de performances des réseaux de neurones sont mentionnés dans le tableau 5.

NNC	RNs	Algo App	Fct Activ	Erreur	Tps App
3	RN3da	Traingd	Tansig	0.0770734 /0.001	172.736012
3	RN3do	Traingd	Logsig	0.0219858 /0.001	51.485863
3	RN3du	Traingd	Purline	0.0576039/0.01	81.752073
3	RN3la	Trainlm	Tansig	0.0219858 /0.001	51.485863
3	RN3lo	Trainlm	Logsig	0.0214034/0.01	133.783070
3	RN3lu	Trainlm	Purline	0.0434045/0.01	4.895414
5	RN5da	Traingd	Tansig	0.0422764 /0.001	212.641642
5	RN5do	Traingd	Logsig	0.0625453/0.01	218.874082
5	RN5du	Traingd	Purline	0.0566066/0.01	129.496871
5	RN5la	Trainlm	Tansig	0.0191969 /0.001	232.583790
5	RN5lo	Trainlm	Logsig	0.0196136/0.01	229.486308
5	RN5lu	Trainlm	Purline	0.0434045/0.01	7.326268
8	RN8da	Traingd	Tansig	0.0459127 /0.001	326.818193
8	RN8do	Traingd	Logsig	0.0537611/0.01	788.159891
8	RN8du	Traingd	Purline	0.0570229/0.01	161.998806
8	RN8la	Trainlm	Tansig	0.0179834 /0.001	1729.413837
8	RN8lo	Trainlm	Logsig	0.0177791/0.01	2836.280246
8	RN8lu	Trainlm	Purline	0.0434045/0.01	10.330043
10	RN10da	Traingd	Tansig	0.0402905 /0.001	386.112024
10	RN10do	Traingd	Logsig	0.0507359/0.01	831.695216
10	RN10la	Trainlm	Tansig	0.0175489 /0.001	2165.193992
10	RN10lo	Trainlm	Logsig	0.0171493/0.01	3984.843191
10	RN10lu	Trainlm	Purline	0.0434045/0.01	12.032794
15	RN15da	Traingd	Tansig	0.0373874 /0.001	581.533880
15	RN15do	Traingd	Logsig	0.0430928/0.01	1006.535844
15	RN15la	Trainlm	Tansig	0.0158089 /0.001	976.146811
15	RN15lo	Trainlm	Logsig	0.0154409/0.01	2891.120455
15	RN15lu	Trainlm	Purline	0.0434045/0.01	19.980377

Tableau 7 : Résultats d'apprentissage avec la base de données mixte.

Les appellations des différents réseaux de neurones conçus sont codées de la manière suivante :



- Nombre de neurones cachés : 3, 5, 8, 10, 15.
- Algorithmes d'apprentissage :
 - « **d** » : Descente du gradient (Traingd).
 - « **l** » : Levenberg-Marquardt (Trainlm).
- Fonctions d'activation :
 - « **a** » : tangente hyperbolique (Tansig) une sortie entre **-1** et **1**.
 - « **o** » : sigmoïde (Logsig) assurant une sortie entre **0** et **1**.
 - « **u** » : linéaire (Pureline).
- *Remarque* : le temps d'apprentissage varie pour le même réseau de neurones, en fonction des capacités du micro-ordinateur utilisé, (nous avons utilisé un *Intel "R" Celeron "R" M CPU 420, RAM de 1Go, et une vitesse de 1.60 GHz*).
- *Résultats d'apprentissage*

Nous avons adopté deux critères de performances, à savoir l'erreur quadratique (*) et le temps d'apprentissage.

a. Critère d'erreur

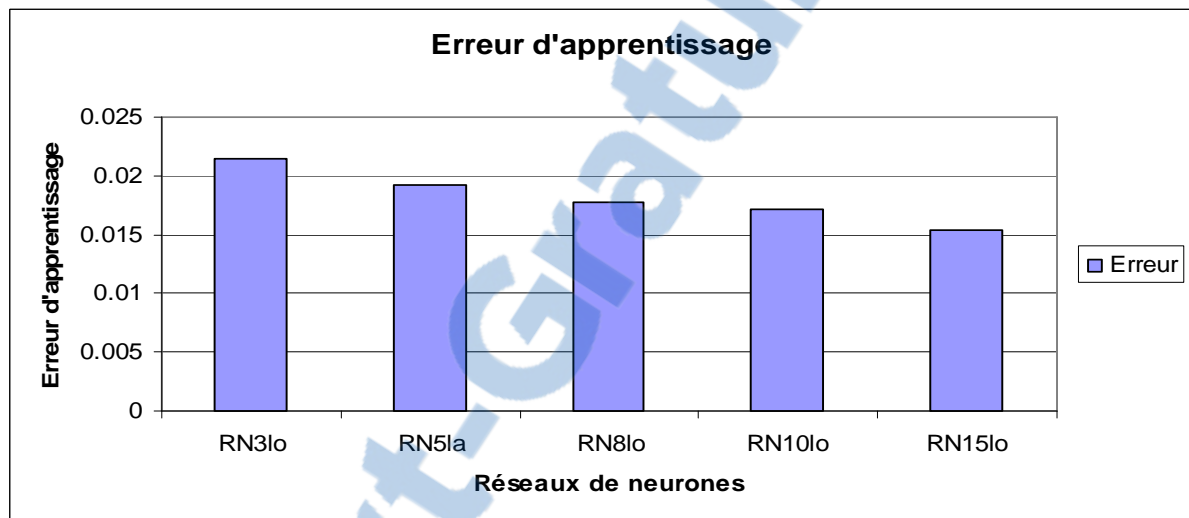
RNs	Nb de NCs	Erreur Quadratique e	Fonction d'activation de la CC	Algorithme d'apprentissage
RN3do	3	0.0219858 /0.001	Sigmoïde	Descente de gradient
RN3la	3	0.0219858 /0.001	Tangente hyperbolique	Levenberg Marquardt
RN3lo	3	0.0214034/0.01	Sigmoïde	Levenberg Marquardt
RN5la	5	0.0191969 /0.001	Tangente hyperbolique	Levenberg Marquardt
RN5lo	5	0.0196136/0.01	Sigmoïde	Levenberg Marquardt
RN8la	8	0.0179834 /0.001	Tangente hyperbolique	Levenberg Marquardt
RN8lo	8	0.0177791/0.01	Sigmoïde	Levenberg Marquardt
RN10la	10	0.0175489 /0.001	Tangente hyperbolique	Levenberg Marquardt

RN10lo	10	0.0171493/0.01	Sigmoïde	Levenberg Marquardt
RN15la	15	0.0158089 /0.001	Tangente hyperbolique	Levenberg Marquardt
RN15lo	15	0.0154409/0.01	Sigmoïde	Levenberg Marquardt

Tableau 8: Les RNs optimaux en valeur d'erreur.

(*) L'erreur quadratique $e = (\text{sortie désirée} - \text{sortie réelle})^2$.

D'après les résultats obtenus, nous remarquons que l'erreur d'apprentissage varie selon l'algorithme d'apprentissage utilisé, la fonction d'activation et le nombre de neurones de la couche cachée.



Histogramme 1 : Erreurs minimums en fonction des différents RNs.

- La fonction d'activation sigmoïde donne des meilleurs résultats avec un nombre assez grand de neurones cachés.
- L'algorithme d'apprentissage de Levenberg Marquardt assure une erreur minimale par rapport à celui de descente de gradient.
- Plus le nombre de neurones cachés augmente, l'erreur diminue le plus.
- Selon l'histogramme 1, le réseau de neurones RN15lo a donné l'erreur quadratique la plus petite $e = 0.0154409/0.01$, avec les paramètres d'apprentissage : (15 neurones cachés, l'algorithme de Levenberg Marquardt et la fonction d'activation de la couche cachée sigmoïde).

b. Critères du temps d'apprentissage

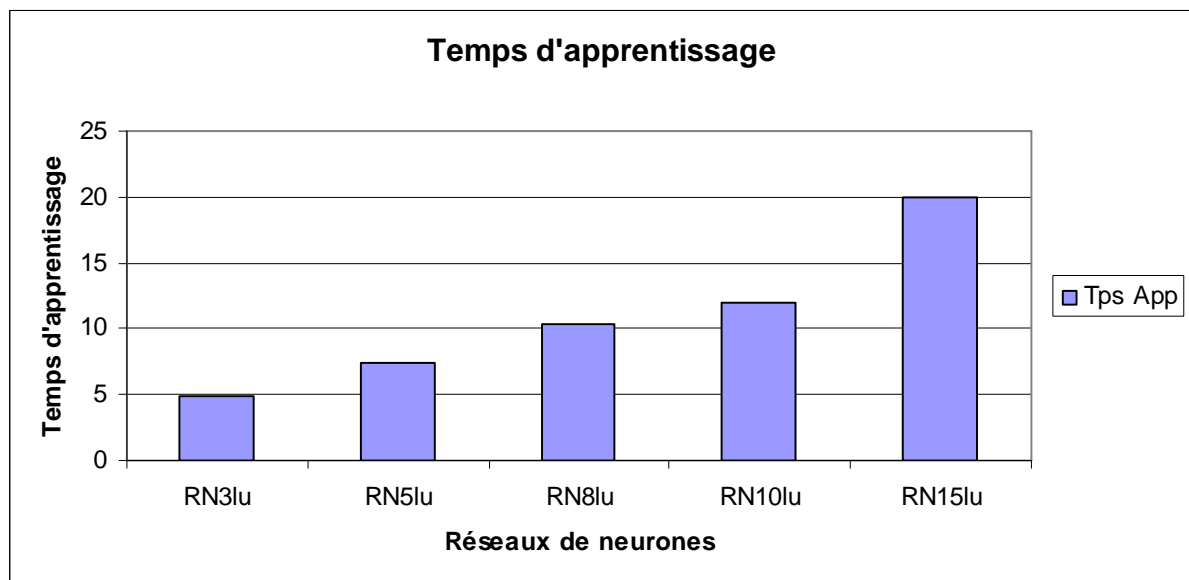
Nous notons que le temps d'apprentissage (t) est le temps nécessaire à la convergence de l'algorithme d'apprentissage.

RNs	Nombre de neurones cachés	Temps d'apprentissage t	Fonction d'activation de la CC	Algorithme d'apprentissage
RN3do	3	51.485863	Sigmoïde	Descente de

				gradient
RN3la	3	51.485863	Tangente hyperbolique	Levenberg Marquardt
RN3lu	3	4.895414	Linéaire	Levenberg Marquardt
RN5lu	5	7.326268	Linéaire	Levenberg Marquardt
RN8da	8	326.818193	Tangente hyperbolique	Descente de gradient
RN8lu	8	10.330043	Linéaire	Levenberg Marquardt
RN8du	8	161.998806	Linéaire	Descente de gradient
RN10da	10	386.112024	Tangente hyperbolique	Descente de gradient
RN10lu	10	12.032794	Linéaire	Levenberg Marquardt
RN15lu	15	19.980377	Linéaire	Levenberg Marquardt

Tableau 9 : Les RNs optimaux en valeur de temps d'apprentissage.

Le tableau 7 comporte les meilleurs réseaux de neurones en tenant en considération la notion du temps d'apprentissage.



Histogramme 2 : Temps d'apprentissage minimums en fonction des différents RNs.

- La fonction d'activation linéaire donne des bons résultats quelque soit le nombre assez grand de neurones cachés.

- L'algorithme d'apprentissage de Levenberg Marquardt assure un temps d'apprentissage minimal par rapport à celui de descente de gradient.
- Contrairement à l'erreur d'apprentissage, le temps d'apprentissage atteint sa valeur minimale avec peu de neurones cachés.
- Selon l'histogramme 2, le réseau de neurones RN3lu a été obtenu en un temps d'apprentissage minimum $t=4.895414$ secondes. avec les paramètres d'apprentissage : (3 neurones cachés, l'algorithme de Levenberg Marquardt et la fonction d'activation de la couche cachée linéaire).

1.2. Apprentissage avec la base des battements normaux "N"

Nous avons réalisé l'apprentissage avec une base d'apprentissage comportant uniquement des battements cardiaques normaux, pour voir l'influence du type des exemples de la base de données utilisée sur l'apprentissage.

RNs	Erreur	Temps d'apprentissage
RN3da	0.000998224/0.001	38.781147
RN10la	4.61072e-005/0.001	5.736833
RN15do	0.000998975/0.001	145.343319
RN15lo	8.59926e-005/0.01	7.503616

Tableau 10: Résultats d'apprentissage en effectuant l'apprentissage avec la base "N".

Le tableau 8 présente les quatre meilleurs réseaux de neurones, pour les différentes expérimentations réalisées.

La figure 3 représente la variation de l'erreur de l'algorithme d'apprentissage du réseau de neurones RN10la, a réalisé les meilleures performances en termes d'erreur et de temps d'apprentissage. Après une seule itération, l'erreur obtenue est $e= 5.11752e-005$, avec un temps d'apprentissage de 6.197462 secondes.

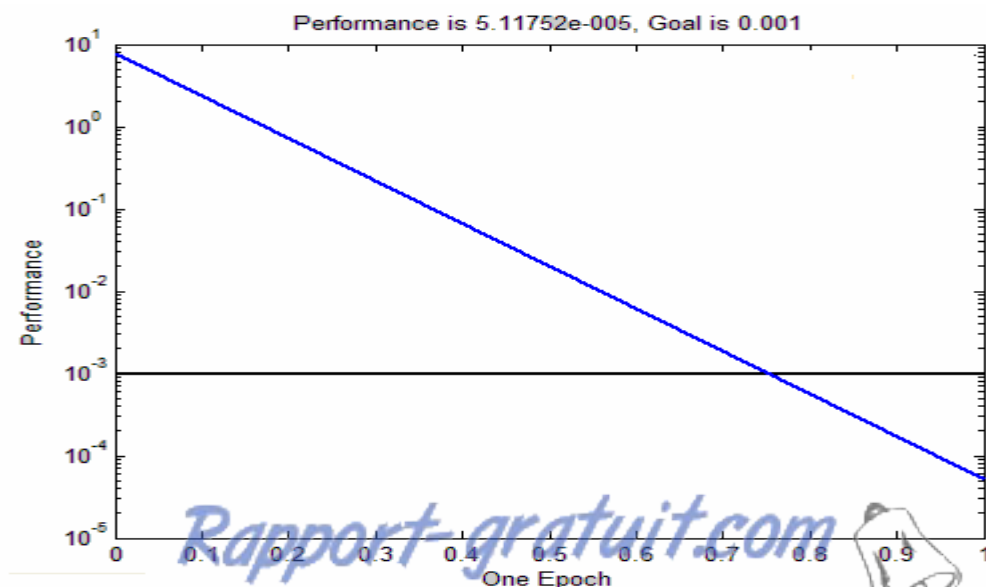
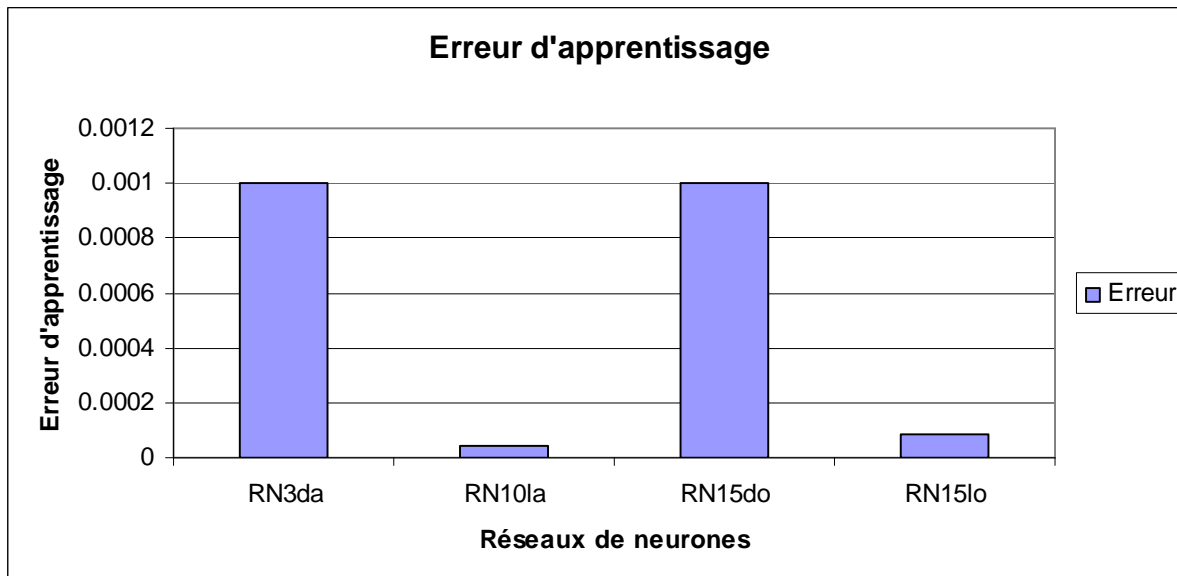


Figure 34 : Variation de l'erreur du RN (RN10la) avec la base des battements "N".

▪ *Résultats d'apprentissage*

a. Critère d'erreur



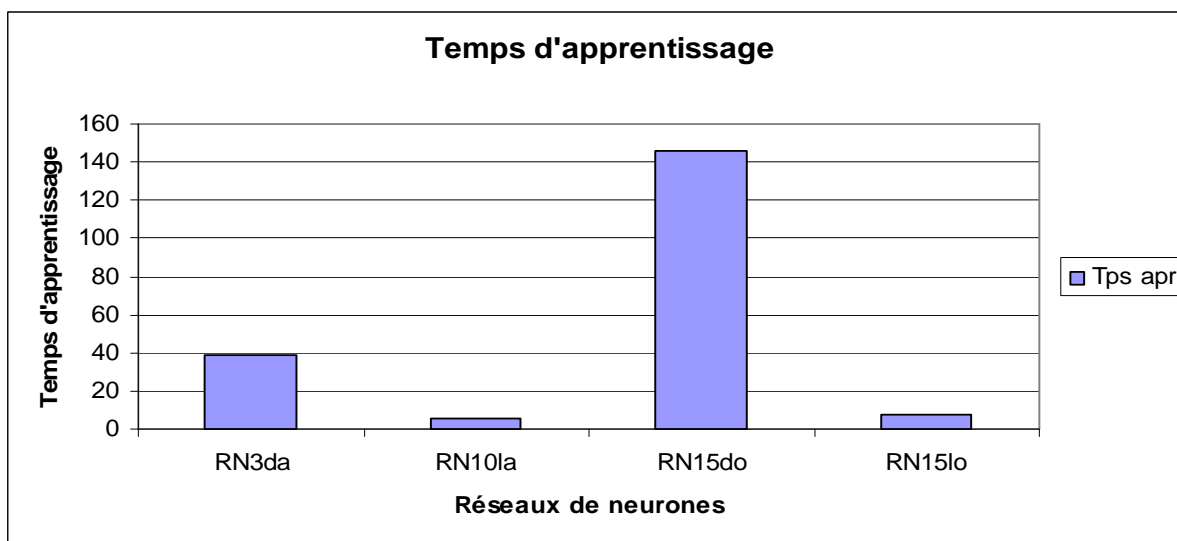
Histogramme 3: Erreurs d'apprentissage avec la base des battements "N".

D'après l'histogramme 3, il est clair que les réseaux de neurones (RN10la et RN15lo) réalisent un apprentissage avec une erreur optimale mais cela en respectant les critères suivants :

- L'utilisation des fonctions d'activation logarithmique et tangente sigmoïde avec un grand nombre de neurones cachés.
- Ainsi que l'algorithme d'apprentissage de Levenberg Marquardt a permis de minimiser l'erreur (histogramme 3).

b. Critère du temps d'apprentissage

Les temps d'apprentissage obtenus avec les réseaux de neurones appris en utilisant la base d'apprentissage des battements "N" sont cités dans le tableau 8.



Histogramme 4 : Temps d'apprentissage avec la base des battements "N".

Les meilleurs temps d'apprentissage ont été établis par les réseaux de neurones utilisant à la fois les fonctions d'activation logarithmique et tangente sigmoïde ainsi que l'algorithme de Levenberg Marquardt, et en faisant varier le nombre de neurones cachés entre **10** et **15**.

1.3. Apprentissage avec la base des battements ESV "V"

Par contre, dans ce cas on n'a utilisé que des battements extrasystoles ventriculaires pour faire l'apprentissage.

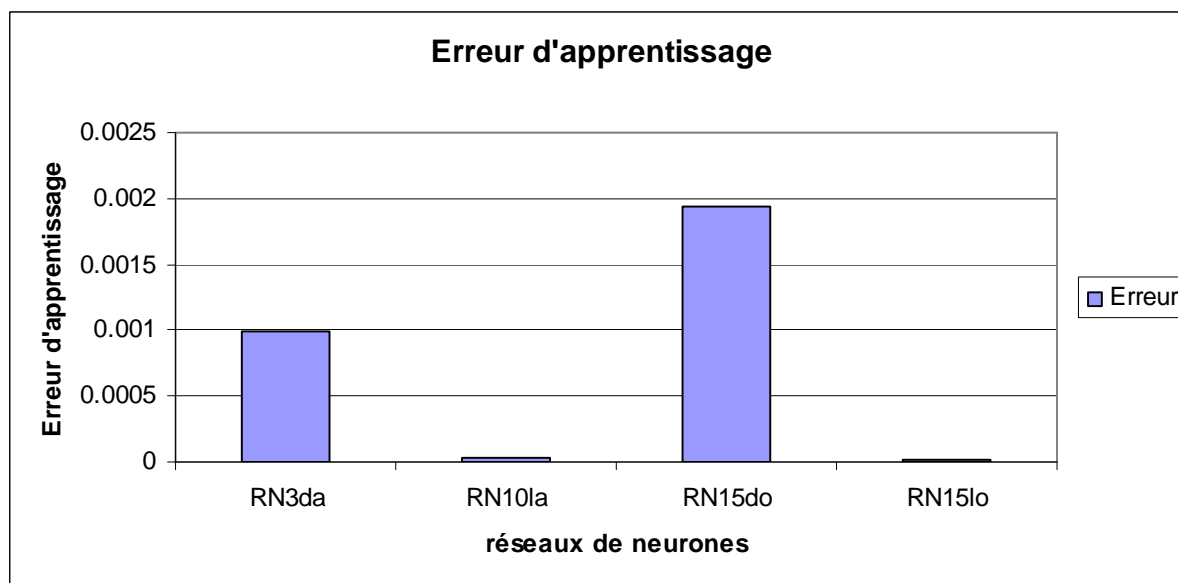
RNs	Erreur	Temps d'apprentissage
RN3da	1.711944	0.000985993/0.001
RN10la	1.766658	3.29587e-005/0.001
RN15do	51.028860	0.00193962/0.001
RN15lo	1.969287	2.02454e-005/0.01

Tableau 11: Résultats obtenus en effectuant l'apprentissage avec la base des battements "V".

- *Résultats d'apprentissage*

- a. Critère d'erreur

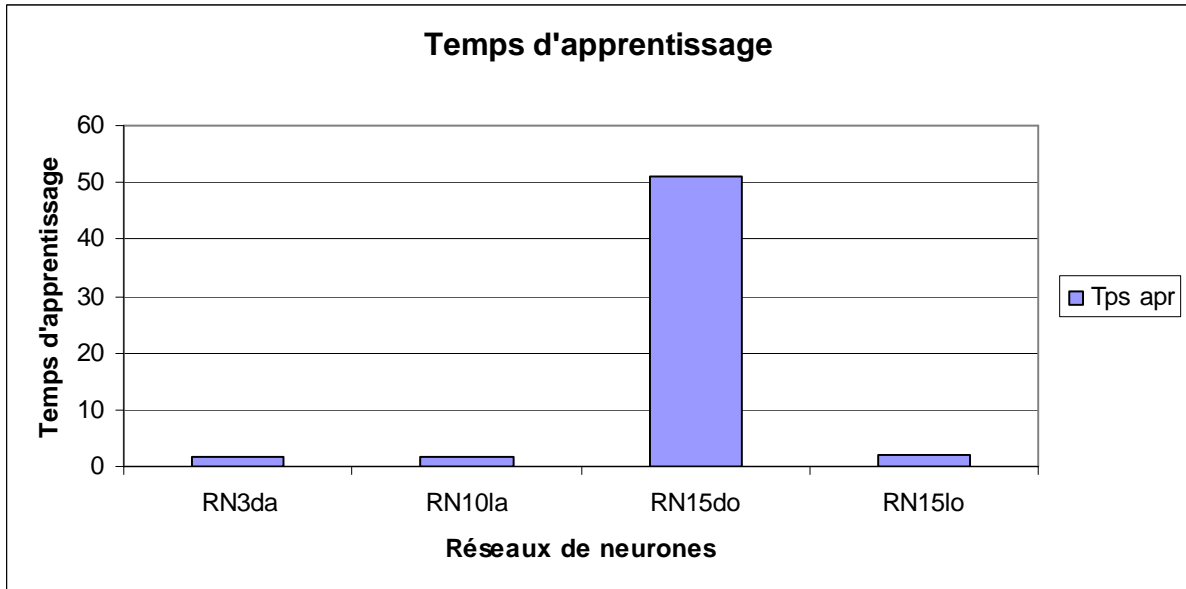
Ce que nous constatons d'après d'histogramme 5 que la meilleure performance a été obtenue par le biais des réseaux de neurones RN10la et RN15lo (histogramme 5).



Histogramme 5 : Erreurs d'apprentissage avec la base des battements "V".

La combinaison des fonctions (sigmoïde et tangente hyperbolique) avec l'algorithme de Levenberg Marquardt minimise l'erreur.

- b. Critère du temps d'apprentissage



Histogramme 6 : Temps d'apprentissage avec la base des battements "V".

Par rapport aux valeurs de l'erreur obtenues, nous remarquons qu'en plus des réseaux de neurones RN10la et RN15lo qui apprennent en temps minimal, en plus le réseau RN3da.

2. Phase de test (validation)

Comme nous sommes intéressés par les battements normaux et les battements ventriculaires prématurés, nous abordons chaque cas indépendamment.

2.1. Evaluation avec la base des battements mixte

Dans cette expérimentation, nous avons utilisé une base de test contenant un mélange de battements "N" et "V" qui n'étaient pas inclus dans la base d'apprentissage.

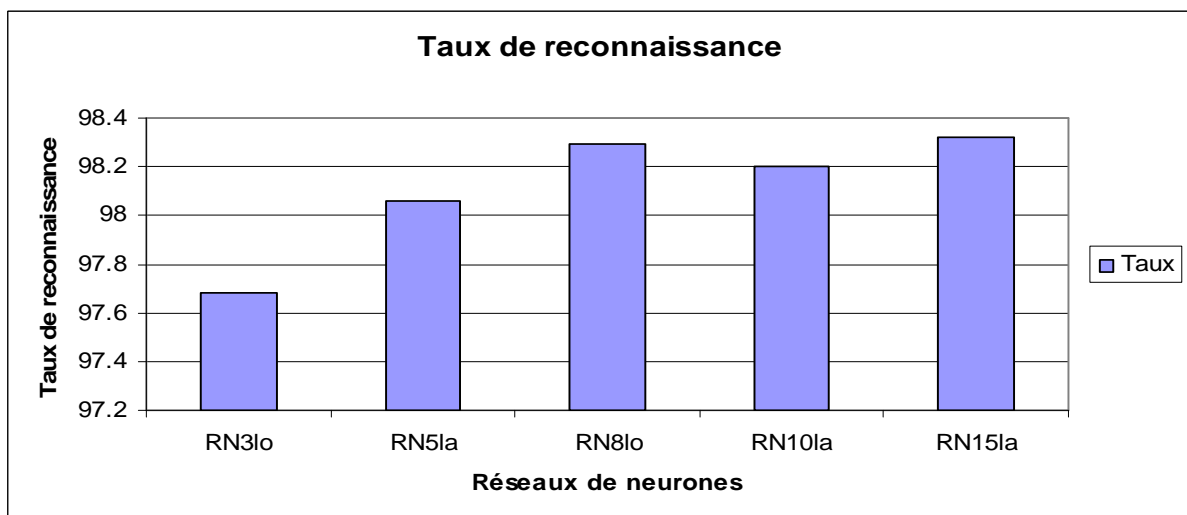
Les valeurs de performance de test sont représentées dans le tableau 9.

Réseaux	Taux (%)	SE (%)
RN3da	91.2292	0
RN3do	91.2507	0.5724
RN3du	93.8612	35.7318
RN3la	97.5186	83.2379
RN3lo	97.6836	86.9174
RN3lu	95.8907	55.7055
RN5da	94.9978	46.9747
RN5do	91.2507	0.4497
RN5du	94.6608	43.7040
RN5la	98.0601	87.8986
RN5lo	97.8665	84.2600
RN5lu	95.8907	55.7055
RN8da	96.0951	66.8029
RN8do	91.8280	8.0540
RN8du	94.2090	38.6754

RN8la	98.0744	88.1030
RN8lo	98.2932	90.0654
RN8lu	95.8907	55.7055
RN10da	96.4931	62.8782
RN10do	93.6460	29.6811
RN10la	98.1999	89.0798
RN10lo	98.1067	87.8168
RN10lu	95.8907	55.7055
RN15da	96.7728	70.8726
RN15do	96.0521	57.2597
RN15la	98.3219	90.4255
RN15lo	98.2645	89.6975
RN15lu	95.8907	55.7055

Tableau 12 : Résultats de test avec la base mixte.

a. Taux de reconnaissance



Histogramme 7 : Taux de reconnaissance des meilleurs RNs (test avec la base mixte).

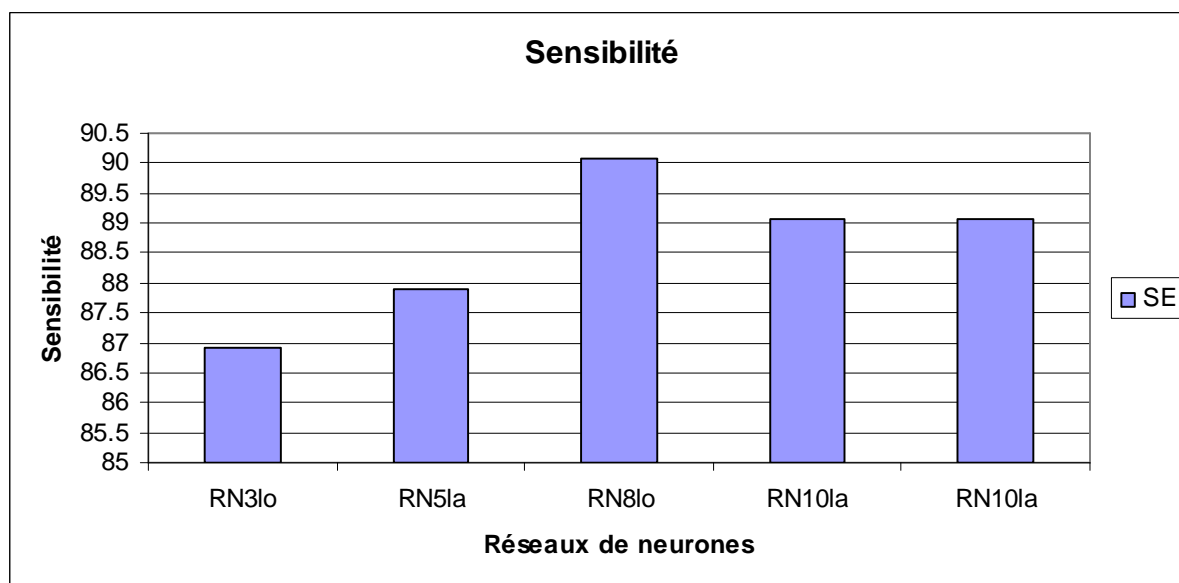
L'histogramme 7 représente les réseaux de neurones (pour chaque nombre de neurones cachés utilisé) qui ont donné des taux de reconnaissance élevés.

- La base de test mixte nous a permis d'avoir des taux de reconnaissance optimaux variant entre 97.6836 % et 98.3219 %.
- Le réseau de neurones qui a assuré le meilleur taux de reconnaissance est celui constitué de 15 neurones cachés avec un apprentissage utilisant l'algorithme de Levenberg Marquardt et la fonction d'activation tangente sigmoïde.
- Les réseaux de neurones RN3da, RN3do, RN5do et RN8do ne donnent pas des résultats satisfaisants, et cela grâce à la mauvaise classification des différents battements est dû probablement au choix des paramètres d'apprentissage, qui sont généralement liés avec le choix du nombre des neurones cachés et les fonctions d'activation.

b. Sensibilité

L'histogramme 8 montre que la sensibilité des réseaux de neurones est meilleure pour le réseau RN8lo (Se=90.0654 %).i.e. le nombre des vrais positifs est élevé relativement aux autres réseaux de neurones.

Nous remarquons que les réseaux de neurones possédant un nombre petit de neurones cachés, ne sont pas assez sensible à la reconnaissance des pathologies. Et par la suite le nombre de neurones dans la couche cachée influe fortement sur la sensibilité du réseau.



Histogramme 8 : Sensibilité des meilleurs RNs (test avec la base mixte).

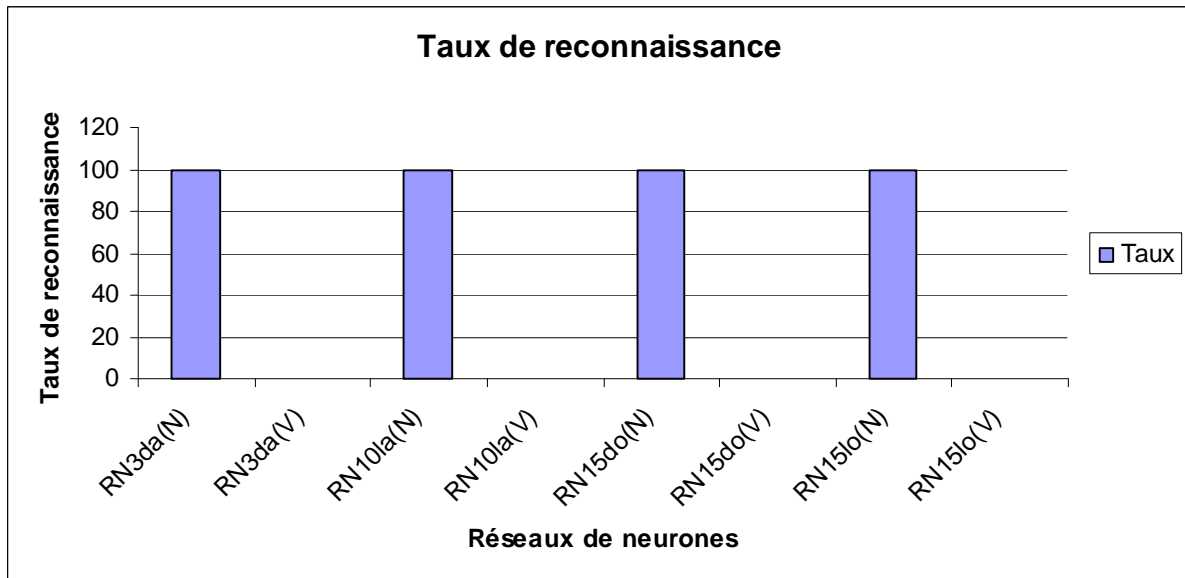
2.2. Evaluation avec la base de test des battements "N"

Pour ces essais, nous avons utilisé une base de test des cas normaux pour tester les réseaux de neurones appris soit avec la base des battements "N" ou la base des battements "V".

RNs	Base d'apprentissage	Base test	Temps de test (secondes)	Taux (%)	Sp (%)
RN3da	N	N	0.198124	100	100
RN3da	V	N	0.195165	0	0
RN10la	N	N	0.088834	100	100
RN10la	V	N	0.090020	0	0
RN15do	N	N	0.120956	100	100
RN15do	V	N	0.273487	0	0
RN15lo	N	N	0.121564	100	100
RN15lo	V	N	0.120272	0	0

Tableau 13 : Evaluation des RNs avec la base de test "N"

a. Taux de reconnaissance

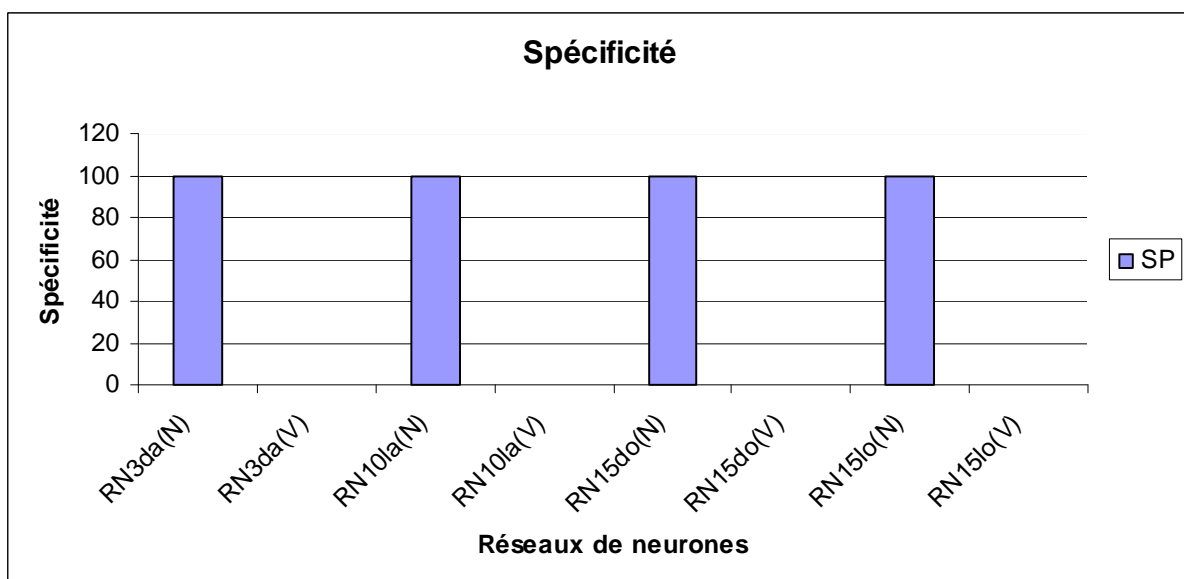


Histogramme 9 : Taux de reconnaissance des RNs (test avec la base "N").

Le résultat obtenu est logique, le taux de reconnaissance est à 100 % pour les réseaux appris avec la base d'apprentissage "N", et vaut zéro (0) pour ceux appris avec la base d'apprentissage "V" et pour cela ces réseaux n'ont reconnu aucune arythmie.

b. Spécificité

La spécificité d'un réseau de neurones représente la fraction des cas réel non visé correctement rejeté.



Histogramme 10 : Spécificité des RNs avec la base de test "N".

2.3. Evaluation avec la base de test des battements “V”

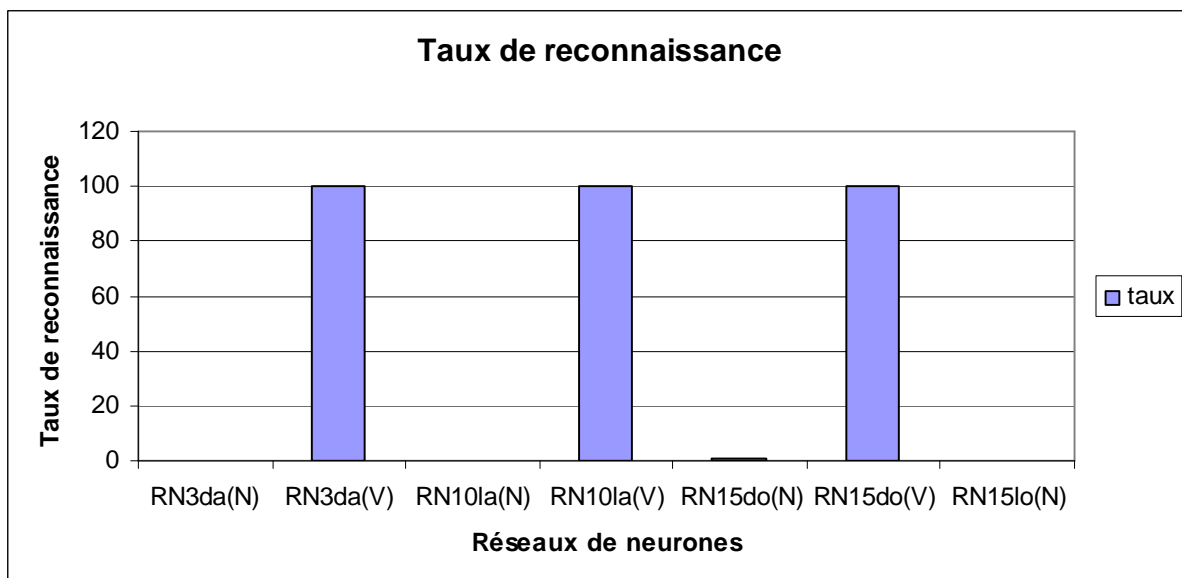
Contrairement à la section précédente, nous utilisons maintenant une base de test “V” avec les réseaux utilisant en phase d’apprentissage soit la base des battements “N”, ou la base des battements “V”.

RNs	Base d’apprentissage	Base test	Temps de test (secondes)	Taux (%)	Se (%)
RN3da	N	V	0.013665	0	0
RN3da	V	V	0.013259	100	100
RN10la	N	V	0.172867	0	0
RN10la	V	V	0.171574	100	100
RN15do	N	V	0.026220	0.8621	0.8621
RN15do	V	V	0.021902	100	100
RN15lo	N	V	0.173380	0	0
RN15lo	V	V	0.018643	100	100

Tableau 14: Evaluation des RNs avec la base de test “V”.

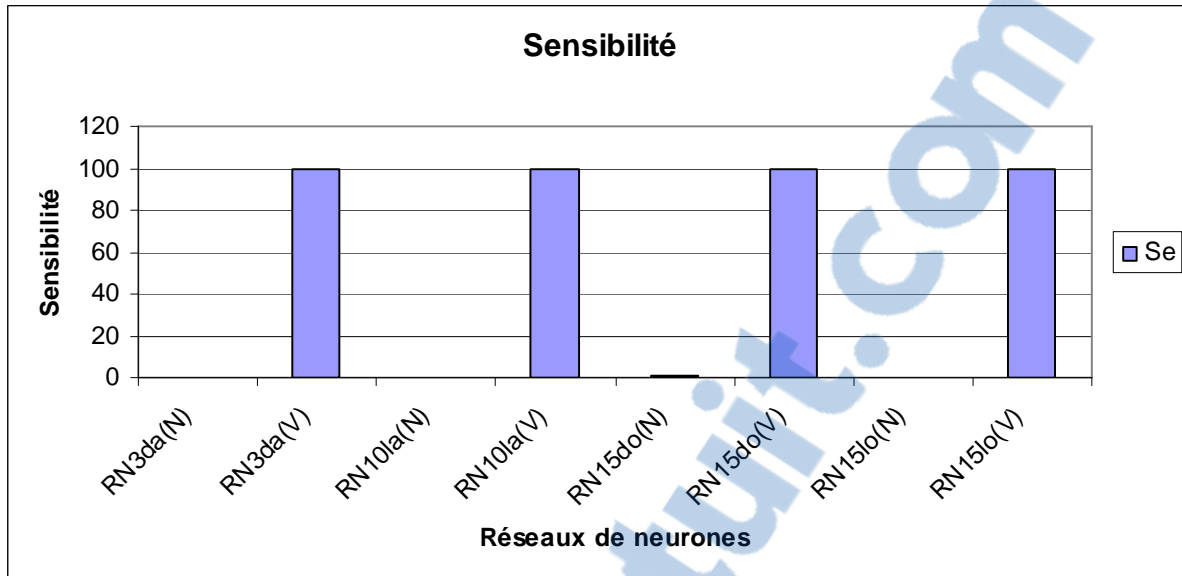
a. Taux de reconnaissance

De même, le test des réseaux de neurones avec la base “V” mène à des résultats parfaits (100%) pour les réseaux appris avec la base d’apprentissage “V”, et des taux de reconnaissance nuls pour les réseaux appris avec la base d’apprentissage “N”, ce qui est réel et satisfaisant, puisqu’il n’y a aucun cas normal dans la base de test, le réseau n’arrive pas à les détecter.



Histogramme 11 : Taux de reconnaissance des RNs (test avec la base “V”).

b. Sensibilité



Histogramme 12 : Sensibilité des RNs avec la base de test "V".

La sensibilité de détection des battements ESV pour les réseaux de neurones appris avec la base "V" est égale à 100 %. Et elle est nulle pour les autres.

- A la fin, nous pouvons dire que l'utilisation d'une base d'apprentissage mixte donne des résultats satisfaisant, même n'atteignant pas les 100 % comme dans les réseaux entraînés avec la même classe (N ou V), mais avec un compromis des performances, ce qui rend les réseaux généralisés plus puissants.
- Donc, nous remarquons que les performances d'un classifieur neuronal dépend en plus des paramètres d'apprentissage (poids, fonctions d'activation,...) de la distribution des exemples dans la base d'apprentissage.

VII. Conclusion

L'objectif de ce chapitre était d'appliquer les principes théoriques des réseaux de neurones afin de réaliser un classifieur neuronal des battements cardiaques.

Ainsi, plusieurs classifieurs à deux classes (battement normal et battement ESV) ont été conçu avec des architectures différentes et une variété de paramètres d'apprentissage à savoir les algorithmes d'apprentissage et les fonctions d'activation et des distributions différentes des exemples dans la base d'apprentissage (N ou V).

Une étude de ces classifieurs a été faite concernant les performances d'apprentissage (erreur et temps d'apprentissage), ainsi qu'une évaluation pour trouver les meilleurs réseaux de neurones, ce qui reste purement empirique.

Enfin, nous notons que la faisabilité d'une amélioration des performances en enrichissant la base d'apprentissage avec plus de battements de différentes classes, et puis en élargissant le vecteur d'entrée avec d'autres paramètres pertinents pour pouvoir augmenter le taux de reconnaissance.

Conclusion générale

Tout au long du présent travail, nous avons tiré parti de deux caractéristiques fondamentales de l'approche neuronale : la parcimonie et l'existence d'algorithmes d'apprentissage universels, au sens où leur mise en oeuvre ne dépend pas de l'application considérée ni de la complexité du réseau soumis à un apprentissage.

Nous avons développé des classifieurs neuronaux différents, comme des outils d'aide au diagnostic médical, pour reconnaître les anomalies cardiaques de type battements ventriculaires prématurés (BVP).

Une étude comparative des différents classifieurs basée sur l'analyse de leurs performances (erreurs quadratiques et temps d'apprentissage) a montré que leurs capacités

sont influencées par le choix des paramètres d'apprentissage (poids et fonctions d'activation), plus la distribution des exemples dans la base d'apprentissage.

Les classifieurs neuronaux se caractérisent par une grande capacité d'apprentissage en termes de précision et d'adaptation (facile à implémenter).

Dans les perspectives de ce travail, nous envisageons l'automatisation de distribution des exemples dans la base d'apprentissage.

Et afin d'améliorer les performances des classifieurs neuronaux, il est nécessaire de faire appel à des techniques d'hybridation à savoir la logique floue et les algorithmes génétiques.

L'hybridation entre deux approches différentes permet de tirer profit des avantages de chacune des méthodes.

La logique floue contribue par sa capacité de représentation des connaissances imprécises. Et augmente le taux d'interprétabilité.

Un modèle hybride de l'approche neuronale et l'approche floue fusionne la capacité de modélisation des réseaux de neurones avec la facilité d'interprétation et d'intégration des connaissances des systèmes flous.

La principale propriété du modèle neuro flou est sa capacité de traiter en même temps des connaissances numériques et symboliques.

Les algorithmes génétiques ont une large capacité d'adaptation. D'où l'apparition des systèmes neuro génétiques pour contourner le problème de choix d'architecture des réseaux de neurones.

Bibliographie

➤ ARTICLES ET THESES

[A. McGinn, 2005] McGinn, S.M., Flesch, T.K., Harper, L.A., and Beauchemin, K.A. (2005). "Whole farm methane emissions using a dispersion model.", 2nd International Conference on Greenhouse Gases and Animal Agriculture, Zurich, Switzerland, September 20-24, 2005.

[Achyara 2004] achayara RU and AL "Classification of cardiac abnormalities using heart rate signals", Med.Bio,Eng.Comp, Vol 42, 2004.

[Adrien Coulet] “ Construction et utilisation d'une Base de Connaissances pharmacogénomique pour l'intégration de données et la découverte de connaissances“, thèse de doctorat, université Henri Poincaré, Nancy 1, 2008.

[AMY 96] Amy DAHAN, “L'essor des mathématiques appliquées aux Etats-unis : l'impact de la Seconde Guerre mondiale“, Revue d'histoire des mathématiques, n°2, 1996.

[Ast 00] K.J. Astrom , B. Wittenmark, Adaptive Control, Addison Wesley, second edition, 2000.

[Benali Radhwane] “Modèle hybride neuro-flou pour la reconnaissance des arythmies cardiaques“, mémoire de magistère, université de Tlemcen, 2008.

[Belgacem 2002] N. Belgacem, M.A Chikh, F. Bereksi Reguig, “ DETECTION ET IDENTIFICATION DES ARYTHMIES CARDIAQUES PAR APPLICATION DES RESEAUX DE NEURONES“, Laboratoire de Génie Biomédical. Département d'électronique. Faculté des Sciences de l'Ingénieur. Université Abou Bekr Belkaïd, 2002. Tlemcen

[Benchaib Yasmine] “Classification neuronale spécialisée des arythmies cardiaques“, mémoire de magistère, université de Tlemcen, 2008.

[Ben MESSAOUD, 2008] Mohamed Ben MESSAOUD, “Traitement des Électrocardiogrammes en Vue de Diagnostic des Pathologies Cardiaques Suivi de Applications des Systèmes Adaptatifs avec Modèle de Référence aux Systèmes Électromécaniques“, L'École Nationale d'Ingénieurs de Sfax, 2008.

[Bishop 95] C. Bishop. Neural networks for pattern recognition. Oxford University Press, 1995.

[Blo] P. Bloxom, The armchair ornithologist, Available on line.

[CHIKH 2005] Mohammed Amine CHIKH, “Analyse du signal ECG par les réseaux de neurones et la logique floue : Application à la reconnaissance des battements ventriculaires prématurés“, université de Tlemcen, 2005.

[DAV 91] Davis, L. (1991), “Bit-Climbing, Representational Bias, and Test Suite Design”, in: [Belew and Booker 1991].

[DeBakey, 1997] M. E. DeBakey and A. M. Gotto, Le coeur en action: Institut d'édition Sanofi~Synthelabo, 1997.

[Einthoven, 1941] W. Einthoven, "The galvanometric registration of the human electrocardiogram, likewise a review of the use of capillary-electrometer in physiology", in Cardiac classics: Willius FW-CV Mosby, St Louis, 1941.

[Eur Heart J, 1998] “Recommandations of a Task Force of the European Society of Cardiology and the European Resuscitation Council on the Pre Hospital Management of Acute Heart Attacks“, 1998.

[François-Benoît VIALATTE] “Aide au diagnostic d’anomalies cardiaques sur enregistrements électrocardiographiques de longue durée (tracés Holter) à l’aide de chaînes de Markov cachées (Hidden Markov Models ou HMM)“, Mémoire pour le DEA de Sciences Cognitives, Université Pierre et Marie Curie, Paris VI, 2002.

[François Portet] “ Pilotage d'algorithmes pour la reconnaissance en ligne d'arythmies cardiaques“, thèse de doctorat, Université de Rennes 1, 2005.

[Gol67] E. Mark Gold, Language Identification in the Limit, Information and Control 10 (1967), no. 5.

[GOL 89] D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley, Massachusetts, 1989.

[GOL 94] D. Goldberg, Algorithmes Génétiques. Addison Wesley, 1994.

[Grù94] Peter Grunwald, Automatic Grammar Induction, Master’s thesis, Center for Mathematics and Computer Science, Amsterdam, 1994.

[Hedeili 2004] N. Hedeili, R.Kaoulal, M.A.Chikh, “Identification de la pathologie cardiaque ESV par un classificateur neuronal“, Laboratoire de Génie Biomédical - G.B.M-, Département d’Electronique, Faculté des Sciences de l’ingénieur, Université de Abou bekr Belkaid, Tlemcen, 2004.

[Hedeili Nawel] “Classification des arythmies cardiaques par l’analyse à composantes principales et les réseaux de neurones“, mémoire de magistère, université de Tlemcen, 2004.

[Hervé Stoppiglia] “Méthodes statistiques de sélection de modèles neuronaux, applications financières et bancaires“, thèse de doctorat de l’université Pierre et Marie Curie Paris VI, 1997.

[HOL 75] Holland, J. (1975), Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Systems, the University of Michigan Press, Ann Arbor, MI.

[HORNİK et al. 1989] K. HORNİK, M. STINCHCOMBE, H. WHITE, "Multilayer Feedforward Networks are Universal Approximators", Neural Networks Vol. 2, 1989.

[HORNİK et al. 1990] Hornik, K., Stinchcombe, M., and White, H. (1990). "Universal Approximation of an Unknown Mapping and its Derivatives Using Multilayer Feedforward Networks," Neural Networks.

[HORNİK 1991] Hornic K., “Approximation capabilities of multi-layer feedforward networks“, IEEE Trans. Neural Networks, 1991.

[Hurst, 1990] J. W. Hurst, Ventricular Electrocardiography: Lippincot Williams & Milkins Publishers, 1990.

[Hussein Atoui] “Conception des systèmes intelligents pour la telemedicine citoyenne“, these de doctorat de l’INSA de Lyon, 2006.

- [J. McMurray et al., 1994] "Cardiology-I: Treatment of myocardial infarction, unstable angina, and angina pectoris", BMJ, Vol. 309, N°6965, 1994.
- [Kohonen 1990] T. Kohonen (T.) 1990, The self-organizing map. Proceedings of the IEEE 9, 1990.
- [Levenberg 44] K. Levenberg, "A method for the solution for certain non linear problems in least squares", Quarterly Journal of Applied Mathematics, vol. 2, 1944.
- [Marey, 1876] E. Marey, "Des variations électriques des muscles et du coeur en particulier, étudiées au moyen de l'électromètre de M. Lippmann", CR Acad Sci (Paris), vol. 82, 1876.
- [Marquardt 63] Marquardt D.W., "An algorithm for least-squares estimation of non-linear parameters", Journal of the society of Industrial and applied Mathematics 11(2), 1963.
- [Mit80] Tom M. Mitchell, The need for biases in learning generalizations, Tech. Report CBM-TR-117, Rutgers University, NJ, Dept of Computer Science, 1980.
- [Mit97] T. Mitchell. Machine Learning. McGraw-Hill, 1997.
- [Mohamed Samir TOUKOUROU] "Application de l'apprentissage artificiel à la prévision des crues éclair", these de doctorat, ECOLE NATIONALE SUPERIEURE DES MINES DE PARIS.
- [Mue95] Axel Mueller, Natural kinds and projectible predicates, Sorites 1 (1995),[http://sowi.iwp.uni-linz.ac.at/Sorites/Natural Kinds.html](http://sowi.iwp.uni-linz.ac.at/Sorites/Natural%20Kinds.html)
- [Nash 90] Nash J.C, "Compact Numerical Methods for Computers: linear algebra and function minimisation", Ed. Adam Hilger, 1990.
- [NERRAND et al. 1991] [NERRAND ET AL. 91] O. NERRAND, P. ROUSSEL-RAGOT, L. PERSONNAZ, G. DREYFUS, S. MARCOS, O. MACCHI ET C. VIGNAT: "Neural Networks Training Schemes for Non-Linear Adaptive Filtering and Modelling". Dans Proc. of the International Joint Conference of Neural Networks. IJCNN-91, Seattle, juin, 1991.
- [Olivier Bousquet] "Apprentissage automatique et simplicité", supervisé par François Yvon, 1999.
- [Pan 85] J. Pan and W. J. Tompkins, "A Real-Time QRS Detection Algorithm", IEEE Transactions on Biomedical Engineering, vol. BME 32, 1985.
- [Powell 76] Powell M.J.D, "Some global convergence properties of a variable metric algorithm for minimization without exact line searches", SIAM-AMS proceedings 9, R.W. Cottle and C.E. Lenke, Eds. Providence RI, London, 1976.
- [Press 92] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, Numerical Recipes in C: Cambridge University Press, 1992.
- [QUI 86] QUINLAN R., "Induction Decision Trees", Machine Learning, Vol(1) n°1, 1986
- [QUI 93]: J. R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA, 1993.

[R. Luepker, 2005] “Delay in acute myocardial infarction: why don't they come to the hospital more quickly and what can we do to reduce delay? “, Am Heart J., Vol. 150, N°3, 2005.

[Rémi DUBOIS] “Application des nouvelles méthodes d'apprentissage à la détection précoce d'anomalies en électrocardiographie“, thèse de doctorat, université Paris 6, 2004.

[RUM 86a] Rumelhart, D., and D. Zipser (1986), “Feature Discovery by Competitive Learning”, in: [Rumelhart, McClelland 1986].

[Rumelhart 86] Rumelhart D.E., McClelland J.L., “Parallel distributed processing. Exploration in the Microstructure of Cognition“, Ma: MIT Press Cambridge 1986.

[RUMELHART et al. 1986] Rumelhart, D.E., Hinton, G.E., Williams, R.J. “Learning internal representations by error propagation“, Ma: MIT Press Cambridge 1986.

[Sekkal Mansouria] “Apprentissage génétique d'un classifieur neuronal, application en cardiologie“, mémoire de magistère, université de Tlemcen, 2009.

[Sim83] Herbert A. Simon, Why should machines learn?, Machine Learning : An artificial intelligence approach (R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, eds.), vol. I, Morgan Kaufmann, 1983.

[Val84] Leslie G. Valiant, A theory of the learnable, Communications of the ACM 27 (1984), no. 11.

[Vap95] Vladimir N. Vapnik, the Nature of Statistical Learning Theory, Springer-Verlag, September 1995.

[Waller, 1893] A. Waller, An introduction to human physiology. New-York: Longmans Green, 1893.

[Wol 92] D. Wolpert, On over fitting avoidance as bias, Tech. Report SFI TR 92-03-5001, Santa Fe Institute, Santa Fe, NM, USA, 1992.

[Wolfe 69]: Wolfe P. “Convergence Conditions for ascent methods“S.I.A.M. Review 11.

[World Health Statistics 2006] “World Health Organization “, Tech.Rep., 2006.

➤ **LIVRES**

- Apprentissage artificiel : concepts et algorithmes, Antoine Cornuéjols, Laurent Miclet et Yves Kodratoff, Edition Eyrolles, 2003.
- Oxford English Dictionary.
- Réseaux de neurones : méthodologie et applications, G. Dreyfus, J.M. Martinez, M. Samuelides, M.B. Gordon, F. Badran, S. Thiria et L. Héroult, Edition Eyrolles, 2004.
- Réseaux de neurones formels pour la modélisation, la commande, et la classification, L. Personnaz et I. Rivals, CNRS éditions, collection Sciences et Techniques de l'Ingénieur, 2003.

- Réseaux de neurones : de la physique à la psychologie, J.-P. Nadal, Armand Colin, 1993.

Annexes

Annexe 1 : Enregistrements des battements de la base de données MIT BIH

Le tableau 1 représente les différents types de battements des patients de la base de données MIT BIH [<http://www.physionet.org/physiobank/database/mitdb>].

Record	N								V	F	O	N	E	P	F	O	Q
	.	L	R	A	a	J	S	V	F	!	e	j	E	P	f	p	Q
100	2239	-	-	33	-	-	-	1	-	-	-	-	-	-	-	-	-
101	1860	-	-	3	-	-	-	-	-	-	-	-	-	-	-	-	2
102	99	-	-	-	-	-	-	4	-	-	-	-	-	2028	56	-	-

103	2082	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-
104	163	-	-	-	-	-	-	2	-	-	-	-	-	1380	666	-	18
105	2526	-	-	-	-	-	-	41	-	-	-	-	-	-	-	-	5
106	1507	-	-	-	-	-	-	520	-	-	-	-	-	-	-	-	-
107	-	-	-	-	-	-	-	59	-	-	-	-	-	2078	-	-	-
108	1739	-	-	4	-	-	-	17	2	-	-	1	-	-	-	11	-
109	-	2492	-	-	-	-	-	38	2	-	-	-	-	-	-	-	-
111	-	2123	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-
112	2537	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-
113	1789	-	-	-	6	-	-	-	-	-	-	-	-	-	-	-	-
114	1820	-	-	10	-	2	-	43	4	-	-	-	-	-	-	-	-
115	1953	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
116	2302	-	-	1	-	-	-	109	-	-	-	-	-	-	-	-	-
117	1534	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-
118	-	-	2166	96	-	-	-	16	-	-	-	-	-	-	-	10	-
119	1543	-	-	-	-	-	-	444	-	-	-	-	-	-	-	-	-
121	1861	-	-	1	-	-	-	1	-	-	-	-	-	-	-	-	-
122	2476	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
123	1515	-	-	-	-	-	-	3	-	-	-	-	-	-	-	-	-
124	-	-	1531	2	-	29	-	47	5	-	-	5	-	-	-	-	-
200	1743	-	-	30	-	-	-	826	2	-	-	-	-	-	-	-	-
201	1625	-	-	30	97	1	-	198	2	-	-	10	-	-	-	37	-
202	2061	-	-	36	19	-	-	19	1	-	-	-	-	-	-	-	-
203	2529	-	-	-	2	-	-	444	1	-	-	-	-	-	-	-	4
205	2571	-	-	3	-	-	-	71	11	-	-	-	-	-	-	-	-
207	-	1457	86	107	-	-	-	105	-	472	-	-	105	-	-	-	-
208	1586	-	-	-	-	-	2	992	373	-	-	-	-	-	-	-	2
209	2621	-	-	383	-	-	-	1	-	-	-	-	-	-	-	-	-
210	2423	-	-	-	22	-	-	194	10	-	-	-	1	-	-	-	-
212	923	-	1825	-	-	-	-	-	-	-	-	-	-	-	-	-	-
213	2641	-	-	25	3	-	-	220	362	-	-	-	-	-	-	-	-
214	-	2003	-	-	-	-	-	256	1	-	-	-	-	-	-	-	2
215	3195	-	-	3	-	-	-	164	1	-	-	-	-	-	-	-	-
217	244	-	-	-	-	-	-	162	-	-	-	-	-	1542	260	-	-
219	2082	-	-	7	-	-	-	64	1	-	-	-	-	-	-	133	-
220	1954	-	-	94	-	-	-	-	-	-	-	-	-	-	-	-	-
221	2031	-	-	-	-	-	-	396	-	-	-	-	-	-	-	-	-
222	2062	-	-	208	-	1	-	-	-	-	-	212	-	-	-	-	-
223	2029	-	-	72	1	-	-	473	14	-	16	-	-	-	-	-	-
228	1688	-	-	3	-	-	-	362	-	-	-	-	-	-	-	-	-
230	2255	-	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-
231	314	-	1254	1	-	-	-	2	-	-	-	-	-	-	-	2	-
232	-	-	397	1382	-	-	-	-	-	-	-	1	-	-	-	-	-
233	2230	-	-	7	-	-	-	831	11	-	-	-	-	-	-	-	-
234	2700	-	-	-	-	-	-	3	-	-	-	-	-	-	-	-	-

Tableau 15 : Enregistrements de la base de données MIT BIH.

Le tableau 2 comporte les significations des symboles utilisés dans la description des battements dans le tableau 1.

symbol	Meaning
. or N	Normal beat
L	Left bundle branch block beat
R	Right bundle branch block beat
A	Atrial premature beat
a	Aberrated atrial premature beat
J	Nodal (junctional) premature beat
S	Supraventricular premature beat
V	Premature ventricular contraction
F	Fusion of ventricular and normal beat
!	Ventricular flutter wave
e	Atrial escape beat
j	Nodal (junctional) escape beat

E	Ventricular escape beat
P	Paced beat
f	Fusion of paced and normal beat
Q	Unclassifiable beat

Tableau 16 : Signification des symboles.

Annexe 2 : Résultats d'apprentissage et de test

Le tableau 3 présente les différents classifieurs neuronaux conçus, avec leurs caractéristiques et paramètres d'apprentissage utilisés ainsi que les résultats obtenus après apprentissage et test avec la base mixte des battements normaux et ESV.

	NNC	Réseaux	Algo App	Fct Activ	Erreur	Tps App	Taux	SE
RN1	3	RN3da	Traingd	Tansig	0.0770734 /0.001	172.736012	91.2292	0
RN11	3	RN3do	Traingd	Logsig	0.0219858 /0.001	51.485863	91.2507	0.5724
RN21	3	RN3du	Traingd	Purline	0.0576039/0.01	81.752073	93.8612	35.7318
RN2	3	RN3la	Trainlm	Tansig	0.0219858 /0.001	51.485863	97.5186	83.2379
RN12	3	RN3lo	Trainlm	Logsig	0.0214034/0.01	133.783070	97.6836	86.9174
RN22	3	RN3lu	Trainlm	Purline	0.0434045/0.01	4.895414	95.8907	55.7055
RN3	5	RN5da	Traingd	Tansig	0.0422764 /0.001	212.641642	94.9978	46.9747
RN13	5	RN5do	Traingd	Logsig	0.0625453/0.01	218.874082	91.2507	0.4497

RN23	5	RN5du	Traingd	Purline	0.0566066/0.01	129.496871	94.6608	43.7040
RN4	5	RN5la	Trainlm	Tansig	0.0191969 /0.001	232.583790	98.0601	87.8986
RN14	5	RN5lo	Trainlm	Logsig	0.0196136/0.01	229.486308	97.8665	84.2600
RN24	5	RN5lu	Trainlm	Purline	0.0434045/0.01	7.326268	95.8907	55.7055
RN5	8	RN8da	Traingd	Tansig	0.0459127 /0.001	326.818193	96.0951	66.8029
RN15	8	RN8do	Traingd	Logsig	0.0537611/0.01	788.159891	91.8280	8.0540
RN25	8	RN8du	Traingd	Purline	0.0570229/0.01	161.998806	94.2090	38.6754
RN6	8	RN8la	Trainlm	Tansig	0.0179834 /0.001	1729.413837	98.0744	88.1030
RN16	8	RN8lo	Trainlm	Logsig	0.0177791/0.01	2836.280246	98.2932	90.0654
RN26	8	RN8lu	Trainlm	Purline	0.0434045/0.01	10.330043	95.8907	55.7055
RN7	10	RN10da	Traingd	Tansig	0.0402905 /0.001	386.112024	96.4931	62.8782
RN17	10	RN10do	Traingd	Logsig	0.0507359/0.01	831.695216	93.6460	29.6811
RN8	10	RN10la	Trainlm	Tansig	0.0175489 /0.001	2165.193992	98.1999	89.0798
RN18	10	RN10lo	Trainlm	Logsig	0.0171493/0.01	3984.843191	98.1067	87.8168
RN28	10	RN10lu	Trainlm	Purline	0.0434045/0.01	12.032794	95.8907	55.7055
RN9	15	RN15da	Traingd	Tansig	0.0373874 /0.001	581.533880	96.7728	70.8726
RN19	15	RN15do	Traingd	Logsig	0.0430928/0.01	1006.535844	96.0521	57.2597
RN10	15	RN15la	Trainlm	Tansig	0.0158089 /0.001	976.146811	98.3219	90.4255
RN20	15	RN15lo	Trainlm	Logsig	0.0154409/0.01	2891.120455	98.2645	89.6975
RN30	15	RN15lu	Trainlm	Purline	0.0434045/0.01	19.980377	95.8907	55.7055

Tableau 17: Différents classifieurs neuronaux réalisés.

Le tableau 4 comprend les résultats obtenus avec quelques classifieurs neuronaux en réalisant l'apprentissage et le test avec des bases de données contenant un seul type de battements, soit normaux (N) soit ESV (V).

	réseaux	B.apr	B.test	Tps apr	Erreur	Tps test	Taux	TP	TN	FP	FN	SE	SP
RN1	RN3da	N	baseitn	38.781147	0.000998224/0.001	0.198124	100	0	25442	0	0	NaN	100
RN1	RN3da	N	baseitv	38.781147	0.000998224/0.001	0.013665	0	0	0	0	2436	0	NaN
RN1	RN3da	N	baseit	38.781147	0.000998224/0.001	0.045132	91.2292	0	25442	0	2446	0	100
RN1	RN3da	V	baseitn	1.711944	0.000985993/0.001	0.195165	0	0	0	25442	0	NaN	0
RN1	RN3da	V	baseitv	1.711944	0.000985993/0.001	0.013259	100	2436	0	0	0	100	NaN
RN1	RN3da	V	baseit	1.711944	0.000985993/0.001	0.044901	8.7708	2446	0	25442	0	100	0
RN8	RN10la	N	baseitn	5.736833	4.61072e-005/0.001	0.088834	100	0	25442	0	0	NaN	100
RN8	RN10la	N	baseitv	5.736833	4.61072e-005/0.001	0.172867	0	0	0	0	2436	0	NaN
RN8	RN10la	N	baseit	5.736833	4.61072e-005/0.001	0.248315	91.2292	0	25442	0	2446	0	100
RN8	RN10la	V	baseitn	1.766658	3.29587e-005/0.001	0.090020	0	0	0	25442	0	NaN	0
RN8	RN10la	V	baseitv	1.766658	3.29587e-005/0.001	0.171574	100	2436	0	0	0	100	NaN
RN8	RN10la	V	baseit	1.766658	3.29587e-005/0.001	0.095988	8.7708	2446	0	25442	0	100	0
RN19	RN15do	N	baseitn	145.343319	0.000998975/0.001	0.120956	100	0	25442	0	0	NaN	100
RN19	RN15do	N	baseitv	145.343319	0.000998975/0.001	0.026220	0.8621	21	0	0	2415	0.8621	NaN
RN19	RN15do	N	baseit	145.343319	0.000998975/0.001	0.129778	91.3045	21	25442	0	2425	0.8585	100
RN19	RN15do	V	baseitn	51.028860	0.00193962/0.001	0.273487	0	0	0	25442	0	NaN	0
RN19	RN15do	V	baseitv	51.028860	0.00193962/0.001	0.021902	100	2436	0	0	0	100	NaN
RN19	RN15do	V	baseit	51.028860	0.00193962/0.001	0.131026	8.7708	2446	0	25442	0	100	0
RN20	RN15lo	N	baseitn	7.503616	8.59926e-005/0.01	0.121564	100	0	25442	0	0	NaN	100
RN20	RN15lo	N	baseitv	7.503616	8.59926e-005/0.01	0.173380	0	0	0	0	2436	0	NaN
RN20	RN15lo	N	baseit	7.503616	8.59926e-005/0.01	0.131044	91.2292	0	25442	0	2446	0	100
RN20	RN15lo	V	baseitn	1.969287	2.02454e-005/0.01	0.120272	0	0	0	25442	0	NaN	0
RN20	RN15lo	V	baseitv	1.969287	2.02454e-005/0.01	0.018643	100	2436	0	0	0	100	NaN
RN20	RN15lo	V	baseit	1.969287	2.02454e-005/0.01	0.284622	8.7708	2446	0	25442	0	100	0

Tableau 4: Résultats obtenus avec les classifieurs particuliers.

