Tables des matières

TABLES DES FIGURES **5**

INTRODUCTION GENERALE **6**

CHAPITRE I GENERALITES SUR LE TRAITEMENT D'IMAGE **8**

I.1. Introduction *8*
I.2. Historique *8*
I.3. Notions de base *8*
I.3.1. Le traitement d'Image 8
I.3.2. L'image 9
I.3.3. Image numérique 9
I.3.4. Caractéristiques de l'image 10
I.4. Système de traitement d'image *12*
I.4.1. Acquisition et numérisation 13
I.4.2. Visualisation 13
I.5. Prétraitement d'image *13*
I.5.1. Modification d'histogramme 13
I.5.2. Réduction de bruit 14
I.5.3. Rehaussement de contraste 15
I.6. Techniques de traitement d'images *17*
I.6.1. Filtrage 17
I.6.2. Segmentation 21
I.6.3. Binarisation 26
I.7. Conclusion *28*

CHAPITRE II LES RESEAUX DE NEURONES **29**

II.1. Introduction *29*
II.2. Historique & définition *29*
II.2.1. Historique 29
II.2.2. Définition 29
II.3. Neurone biologique *30*
II.3.1. Le neurone 30
II.3.2. Le fonctionnement 31
II.4. Neurone formel *31*
II.4.1. Synoptique 31
II.4.2. Le fonctionnement 32
II.5. Architecture des réseaux de neurones *33*
II.5.1. Réseaux multicouches 33
II.5.2. Réseaux à connexions locales 34
II.5.3. Les réseaux de neurones récurrents 34
II.5.4. Les réseaux de neurones à connexion complète 34
II.6. Apprentissage des réseaux de neurones *34*
II.6.1. Définition 34
II.6.2. Techniques d'apprentissage 34
II.7. Apprentissage dans quelques réseaux célèbres *35*
II.7.1. Perceptron 35
II.7.2. Réseaux multicouches 36
II.7.3. ADALINE 37
II.7.4. Les réseaux de HOPFIELD 38
II.7.5. Les réseaux de KOHONEN 39
II.7.6. Les réseaux RBF 40
II.8. Les étapes de la conception d'un réseau *40*
II.8.1. Choix des échantillons 40
II.8.2. Elaboration de la structure du réseau 40
II.8.3. Apprentissage 40
II.8.4. Validation et tests 41

II.9. Les avantages et les inconvénients *41*
II.10. Conclusion *41*

CHAPITRE III AMELIORATION DE L'IMAGE PAR LES RESEAUX DE NEURONES 42

III.1. Introduction *42*
III.2. Intérêt des réseaux de neurones dans le traitement d'image *42*
III.3. Le principe utilisé *42*
III.4. Architecture proposée du réseau de neurones *43*
III.4.1. Architecture du réseau 43
III.4.2. Parcours de l'image 44
III.4.2.1. Le parcours automatique *44*
III.4.2.2. Le parcours aléatoire *45*
III.4.3. Apprentissage 45
III.4.4. Tests du réseau de neurones 46
III.5. Conclusion *46*

CHAPITRE IV IMPLEMENTATION & SIMULATION 48

IV.1. Introduction *48*
IV.2. Environnement de développement *48*
IV.3. Description de l'application *49*
IV.3.1. Partie traitement d'image 50
IV.3.2. Traitement par réseaux de neurones 54
IV.3.3. Phase des tests & résultats 56
IV.3.4. Conclusion 57

CONCLUSION GENERALE ET PERSPECTIVES 58

REFERENCES BIBLIOGRAPHIQUES 59

Tables des figures

Figure I. 1 : Représentation matricielle d'une image numérique ----------10
Figure I. 2 : Voisinage à 4 ----------11
Figure I. 3: Voisinage à 8 ----------11
Figure I. 4 : Composition d'un système de traitement numérique----------12
Figure I. 5 : Schématisation de la transformation par expansion ----------14
Figure I. 6: Opérateur de SOBEL. ----------24
Figure I. 7 : Opérateur de LAPLACIEN. ----------24
Figure I. 8 : Opérateur de KIRSH. ----------25
Figure I. 9 : Seuillage d'une image. ----------25
Figure II. 1 : le neurone biologique.----------30
Figure II. 2 : Connexion entre les neurones. ----------31
Figure II. 3 : fonctionnement du neurone formel----------32
Figure II. 4 : Neurone formel.----------32
Figure II. 5 : fonction Heaviside. ----------33
Figure II. 6 : fonction signe. ----------33
Figure II. 7 : fonction sigmoïde. ----------33
Figure II. 8 : Réseau de neurone non bouclé. ----------33
Figure II. 9 : Réseau à connexions locales.----------34
Figure II. 10: Réseau de neurone bouclé.----------34
Figure II. 11: Le perceptron avec seuil----------35
Figure II. 12: Le perceptron avec entrée supplémentaire ----------35
Figure III. 1: Configuration classique d'un réseau de neurones. ----------43
Figure III. 2: Architecture d'un filtre neuronal. ----------44
Figure III. 3: Parcours de l'image.----------44
Figure IV. 1: Principe de fonctionnement. ----------49

Introduction Générale

Contexte général

A la fin des années cinquante, de nombreuses recherches ont été effectuées sur le traitement des images spatiales et biomédicales par ordinateur, et avec la naissance des machines de calcul de plus en plus puissantes, une nouvelle discipline a vu le jour « la vision par ordinateur » ou « vision artificielle ».

Il existe différentes techniques pour améliorer la qualité des images. En effet, en plus des méthodes de filtrage et de traitements classiques, il y a une technique qui consiste à utiliser l'image comme un exemple ou un ensemble d'exemples qui servent d'entrée à un réseau de neurones artificiels.

Problématique et objectifs

L'intérêt principal des réseaux de neurones étant leur capacité d'apprentissage et de généralisation. Leur architecture les rendant très intéressants à plusieurs égards : la possibilité de leur implantation sur machine neuronale, leur résistance aux pannes lors du traitement d'images en temps réel. Le traitement d'images est une des applications majeures des réseaux de neurones. Lors de la réalisation des différentes applications de traitement d'image, plusieurs types d'architectures de réseaux et différentes méthodes d'apprentissage ont été utilisés.

L'objet essentiel de notre projet est donc d'utiliser les avantages des réseaux de neurones pour effectuer des traitements locaux sur des images. La couche d'entrée recevra les valeurs des pixels sur un voisinage d'un pixel à traiter, et la couche de sortie sera composée d'un seul neurone produisant le résultat du traitement considéré pour ce pixel.

Dans un premier temps, et après avoir défini un réseau de neurones avec une architecture et des structures de données appropriées, nous l'avons testé sur des images bruitées, puis nous avons expérimenté notre outil pour l'élimination du bruit. Afin d'en parvenir, et dans le but de tester notre modèle à base de réseaux de neurones, nous avons exploité l'habilité de ce type de réseaux dans l'apprentissage supervisé. En effet, nous avons pris une image nette sur laquelle nous avons rajouté du bruit de façon aléatoire. Les données d'apprentissage sont donc l'image bruitée en entrée et l'image d'origine en sortie (et c'est le résultat désiré).

Organisation du mémoire

Ce rapport est décomposé en quatre chapitres :

Le première chapitre : Consacré, aux généralités de traitements d'images, Décrit en premier lieux quelques méthodes de prétraitement d'une image (la modification d'histogramme, la réduction du bruit,

le rehaussement de contraste) et quelques définitions utiles dans ce domaine telles que les notions d'image, pixel, histogramme, voisinage…,

En deuxième lieux, nous avons détaillé le domaine des réseaux de neurones, en présentant les différents aspects ainsi que les différentes méthodes et algorithme d'apprentissage

Le chapitre qui suit, recouvre l'essentiel de notre application, et comprend, l'approche neuronale en détail : le principe de fonctionnement, les différents types d'architecture, domaines d'application. L'architecture de notre réseau de neurones adopté, les méthodes utilisées pour l'extraction des exemples à partir d'une image durant l'apprentissage. Nous y exposons également l'algorithme d'apprentissage de notre réseau.

Le troisième chapitre, présente la méthodologie d'implémentation et les modules de notre application : Environnement de programmation, une description fonctionnelle du logiciel, ainsi les résultats de tests sur plusieurs images.

Nous clôturons enfin notre présente étude par une conclusion générale dans laquelle nous exhibons la puissance des réseaux de neurones dans l'amélioration de la qualité de l'image.

Chapitre I Généralités sur le traitement d'image

I.1. Introduction

Depuis longtemps, le public sait que: « Une image vaut mieux que mille mots », combinée avec la parole, l'image constitue un moyen essentiel dans la communication homme-machine « Une vidéo vaut mille phrases ». De ce fait, le traitement d'image est devenu une discipline nécessaire, mettant en place un ensemble de techniques permettant d'extraire des informations, de modifier une image numérique (dans le but de l'améliorer), et d'automatiser son traitement. En résumé cette discipline permet d' « Apprendre à voir aux machines ».

Le domaine de traitement d'image a vigoureusement évolué, et ses techniques sont actuellement utilisées pour résoudre une variété de problèmes pour lesquels on adapte des solutions selon la nature, les situations et les objectifs à atteindre.

Ce chapitre sera divisé en deux axes, le primer s'intéresse aux prétraitements d'images (notions de base), et le second aux techniques de traitement de ces dernières (filtrage, segmentation, …).

I.2. Historique

Le traitement d'image commence à être étudié vers les années 1920, pour la transmission d'images par le câble sous-marin allant de New York à Londres. En effet, la première numérisation d'image avec compression de données pour envoyer des fax de Londres à New York[1]. Les années 1950, les premiers travaux sur l'analyse d'images dans les chambres à bulles. Les années 1960, ont vu le véritable départ du traitement d'images quand les ordinateurs commencent à être suffisamment puissants pour travailler sur des images. Les premiers essais sur l'extraction automatique d'information ont commencé avec le début de la décennie 70. Apres les années 80, il y a eu un passage de l'image 2D aux modèles 3D.

I.3. Notions de base

I.3.1. Le traitement d'Image

Le traitement d'image est l'ensemble d'opérations qui permettent l'amélioration (filtrage, rehaussement de contraste), la modification (rotation, symétrie, …) et l'extraction de l'information à partir des images (segmentation).

D'un point de vue syntaxique, les opérations de traitement d'images se scindent en trois catégories :

- Les opérations ponctuelles : l'image d'un pixel donné n'est influencée que par la valeur de ce dernier. *Exemple* : *le négatif d'une image*, *l'égalisation de l'histogramme*…
- Les opérations locales : l'image d'un pixel donné dépend de la valeur du pixel en question et de son voisinage. *Exemple* : *le filtrage*…
- Les opérations globales : l'image d'un pixel donné dépend de touts les pixels de la matrice qui représente l'image. *Exemple* : *la transformée de Fourier d'une image.*

[1] http://fr.wikipedia.org/wiki/Traitement_d%27images

I.3.2. L'image

Une image est une représentation planaire d'une scène ou d'un objet situé en général dans un espace tridimensionnel, elle est issue du contact des rayons lumineux provenant des objets formants la scène avec un capteur (caméra, scanner, rayons X, …). Il ne s'agit en réalité que d'une représentation spatiale de la lumière.

L'image est considérée comme un ensemble de points auquel est affecté une grandeur physique (luminance, couleur). Ces grandeurs peuvent être continues (image analogique) ou bien discrètes (images digitales).

Mathématiquement, l'image représente une fonction continue **IF**, appelée fonction image, de deux variables spatiales représentée par **IF**(x, y) mesurant la nuance du niveau de gris de l'image aux coordonnées (x, y).

La fonction Image peut se représenter sous la forme suivante :

$$\begin{aligned} IF : R^2 &\longrightarrow R \\ (x, y) &\longrightarrow IF(x, y) \end{aligned}$$

Avec R : ensemble des réelles.
x et y : Deux variables réelles.

I.3.3. Image numérique

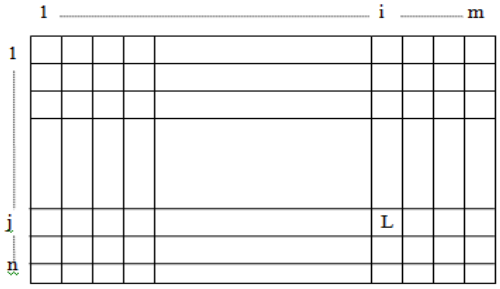
Pour faire un traitement sur une image avec l'outil informatique, il faut quelle soit numériser. L'image numérique est l'image dont la surface est divisée en éléments de taille fixe appelés cellules ou pixels. La numérisation d'une image est la conversion de celle-ci en une image numérique (c'est-à-dire à partir d'une image à distribution continue d'intensité, vers une image numérique) représentée par une matrice bidimensionnelle à valeur numérique f(x, y) où x, y correspondent aux coordonnées cartésiennes (discrètes) d'un point de l'image, et f(x, y) la valeur de ce point. [**Hadjila & Bouabdellah, 03**]

Les différents types de format d'image :

- Image couleur RVB : L'œil humain analyse la couleur à l'aide de trois types de cellules photo réceptrices *'les cônes'*. Ces cellules sont sensibles aux basses, moyennes, ou hautes fréquences (rouge, vert, bleu).
  Pour représenter la couleur d'un pixel, il faut donc donner trois nombres, qui correspondent au dosage de trois couleurs de base : Rouge, Vert, Bleu. On peut ainsi représenter une image couleur par trois matrices chacune correspondant à une couleur de base.
- Image d'intensités : C'est une matrice dans laquelle chaque élément est un réel compris entre 0 (noir) et 1 (blanc). On parle aussi d'image en niveaux de gris, car les valeurs comprises entre 0 et 1 représentent les différents niveaux de gris.
- Image binaire : Une image binaire est une matrice rectangulaire dans l'élément valent 0 ou 1. Lorsque l'on visualise une telle image, les 0 sont représentés par du noir et les 1 par du blanc. [**Naciri & Chaoui, 03**]

Une image numérique est une image f(x, y) discrétisée en coordonnées et en niveaux de gris. Elle peut être considérée comme étant une matrice dont les lignes et les colonnes fournissent les coordonnées, et les valeurs des éléments de la matrice sont les valeurs de niveaux de gris comme le montre la figure suivante (**Figure I. 1**).

Rapport-Gratuit.com



L: Le niveau de gris au point (x, y).
I (i, j)=L.
n : La hauteur de l'image.
m: La largeur de l'image.

Figure I. 1 : *Représentation matricielle d'une image numérique*

I.3.4. Caractéristiques de l'image

L'image est un ensemble structuré d'information caractérisé par les paramètres suivants :

I.3.4.1. Pixel

Le pixel est l'abréviation du mot « Picture élément », est une unité de surface permettant de définir la base d'une image numérique. Il matérialise un point donné (x, y) du plan de l'image. L'information représentée par le pixel est le niveau de gris (ou la couleur) prélevée à l'emplacement correspondant dans l'image réelle. La différence entre image monochrome et image couleur réside dans la quantité d'informations contenue dans chaque pixel, par exemple dans une image couleur (RVB : Rouge, Vert, Bleu) la valeur d'un pixel est représentée sur trois octets pour chaque couleur.

I.3.4.2. Dimension & Résolution

La dimension est la taille de l'image. Elle se présente sous forme d'une matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image.

Par contre, la résolution est la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production d'images. Sur les moniteurs d'ordinateur, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre). On utilise aussi le mot résolution pour désigner le nombre total de pixels horizontaux et verticaux sur un moniteur. Plus ce nombre est grand, plus la résolution est meilleure.

I.3.4.3. Voisinage

Le plan de l'image est divisé en termes de formes rectangulaires ou hexagonales permettant ainsi l'exploitation de la notion de voisinage [**Hadjila & Bouabdellah, 03**]. Le voisinage d'un pixel est formé par l'ensemble des pixels qui se situent autours de ce même pixel. On définit aussi l'assiette comme étant l'ensemble de pixels définissant le voisinage pris en compte autour d'un pixel. [**Naciri & Chaoui, 03**]

On distingue deux types de voisinage :

- Voisinage à 4 : On ne prend en considération que les pixels qui ont un coté commun avec le pixel considéré (**Erreur ! Source du renvoi introuvable.**).
- Voisinage à 8 : On prend en compte tous les pixels qui ont au moins un point en liaison avec le pixel considéré (**Erreur ! Source du renvoi introuvable.**).

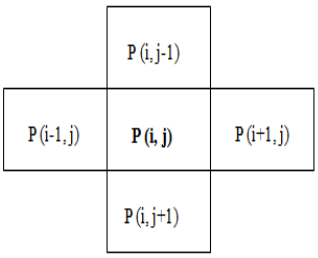


Figure I. 2 : *Voisinage à 4*

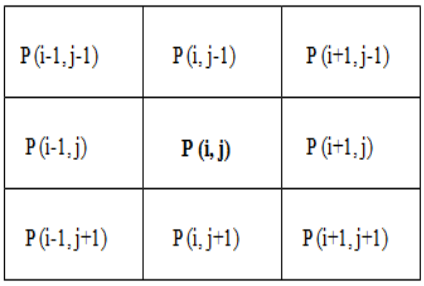


Figure I. 3: *Voisinage à 8*

I.3.4.4. Niveau de gris

C'est la valeur d'intensité lumineuse d'un pixel. Cette valeur peut aller du noir (0) jusqu'au blanc (255) en passant par les nuances qui sont contenues dans l'intervalle [0, 255]. Elle correspond en fait à la quantité de la lumière réfléchie.

Pour 8 bits, on dispose de 256 niveaux de gris dont 40 sont reconnus à l'œil nue. Plus le nombre de bit est grand plus les niveaux sont nombreux et plus la représentation est fidèle. [**Hadjila & Bouabdellah, 03**]

I.3.4.5. Contraste

C'est l'opposition marquée entre deux régions d'une image. Une image contrastée présente une bonne dynamique de la distribution des valeurs de gris sur tout l'intervalle des valeurs possibles, avec des blancs bien clairs et des noirs profonds. Au contraire une image peu contrastée a une faible dynamique, la plupart des pixels ayant des valeurs de gris très proches.

Si *L1* et *L2* sont les degrés de luminosité respectivement de deux zones voisines *A1* et *A2* d'une image, le contraste est défini par le rapport : $C = \frac{L1-L2}{L1+L2}$

I.3.4.6. Luminance [Tabari & Tagma, 96]

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet.

Une bonne luminance se caractérise par :

- Des images lumineuses (brillantes);
- Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir; ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.
- L'absence de parasites.

I.3.4.7. Bruit

Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur [**Gonzales & Wintz, 77**]. C'est un parasite qui représente certains défauts (poussière, petits nuages, baisse momentanée de l'intensité électrique sur les capteurs, …etc.). Il se traduit par des taches de faible dimension et dont la distribution sur l'image est aléatoire.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

I.3.4.8. Contour

Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentant une différence significative. [**Hadjila & Bouabdellah, 03**]

Dans une image numérique, les contours se situent entre les pixels appartenant à des régions ayant des intensités moyennes différentes; il s'agit de contours de type « saut d'amplitude ». Un contour peut également correspondre à une variation locale d'intensité présentant un maximum ou un minimum; il s'agit alors de contour « en toit ». [**Cocquerez & Philipp, 98**]

I.3.4.9. Histogramme de l'image

L'histogramme est un vecteur dont la dimension est égale à 256 (dans le cas d'une image à niveau de gris). Chaque élément de ce vecteur (noté h(i)) représente le nombre de pixels de l'image possédant le niveau de gris i (sa fréquence d'apparition), on peut donc assimiler l'histogramme à la densité de probabilité des intensités lumineuses.

Puisque le nombre des pixels est généralement assez grand, on peut alors normaliser l'effectif de chaque niveau de gris en divisant chaque terme du tableau par la surface du plan exprimée en nombre total des pixels de l'image. De ce fait, chaque case h(i) du vecteur représente la probabilité d'avoir l'intensité i.

Il peut être utilisé pour améliorer la qualité d'image (rehaussement d'image) en introduisant quelques modifications, pour pouvoir extraire les informations utiles de celle-ci.

I.4. Système de traitement d'image

Dans le contexte de la vision artificielle, le traitement d'images se place après les étapes d'acquisition et de numérisation, assurant les transformations d'images et la partie de calcul permettant d'aller vers une interprétation des images traitées. Cette phase d'interprétation est d'ailleurs, de plus en plus intégrée dans le traitement d'images, en faisant appel notamment à l'intelligence artificielle pour manipuler des connaissances, principalement sur les informations dont on dispose à propos de ce que représentent les images traitées (*connaissance du domaine*).

Un système de traitement d'image est généralement composé des unités suivantes :

- Un système d'acquisition et de numérisation qui permet d'effectuer l'échantillonnage et la quantification d'une image.
- Une mémoire de masse pour stocker les images numérisées.
- Un système de visualisation.
- Une unité centrale permettant d'effectuer les différentes opérations de traitement d'images.

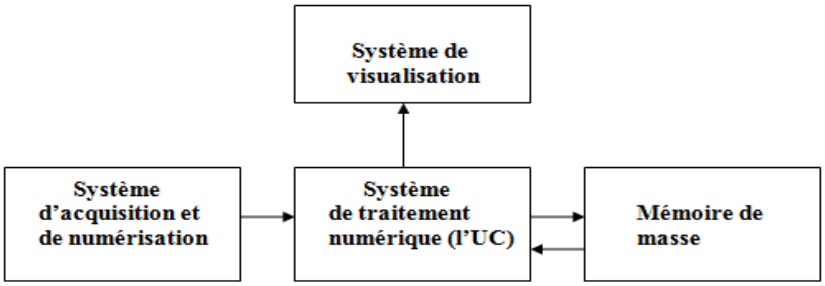


Figure I. 4 : *Composition d'un système de traitement numérique*

I.4.1. Acquisition et numérisation

C'est le mécanisme qui permet l'obtention d'une image numérique (représentée par une matrice) à deux dimensions à partir d'une scène à trois dimensions, en passant par un système optique. Cette opération peut être statique (Appareil Photo, Scanner…etc.) ou dynamique (Caméra, Webcam), dans ce cas on aura une séquence vidéo. L'image continue f(x, y) est approximée par des échantillons qui sont obtenus par discrétisation des coordonnés (x, y) (ce qu'on appelle l'échantillonnage) et la discrétisation des amplitudes de ces points (c'est la quantification). A ce niveau on obtient une donnée brute qui nécessitera d'éventuels traitements pour les phases à venir. La représentation obtenue ne peut être parfaite à cause du bruit introduit dans l'image lors de son acquisition. [**Besançon, 98**]

I.4.2. Visualisation

Un dispositif de visualisation permet l'affichage de l'image. Les restituteurs permettent de transformer le signal numérique qui est la matrice image en un signal analogique visible à l'œil humain. Pour cela, on dispose d'une multitude de supports employés à cet effet (moniteur vidéo, impression sur papier…). [**Hadjila & Bouabdellah, 03**]

I.5. Prétraitement d'image

L'étape de prétraitement a pour but de faciliter la segmentation en renforçant la ressemblance entre pixels appartenant à une même région, ou en accentuant la dissemblance entre pixels appartenant à des régions différentes.

Les images comportent généralement une grande quantité de données. Pour éviter des temps de traitement trop élevés, on cherche donc à leur appliquer des opérateurs locaux, qui ne font appel, pour le calcul, qu'à un nombre limité de pixels, situés dans un proche voisinage du pixel courant. [**Cocquerez & Philipp, 98**]

Quelques méthodes essentielles de prétraitement sont présentées. Elles concernent la modification d'histogramme, la réduction de bruit et le rehaussement de contraste.

I.5.1. Modification d'histogramme

On cherche à améliorer l'image en lui appliquant une transformation ponctuelle d'intensité. A tout pixel d'intensité $\boldsymbol{a_s}$ on associe une intensité $\boldsymbol{a_s = T(a_s)}$. La transformation **T** est choisie croissante, de façon à conserver les contrastes relatifs entre régions (une région claire sur un fond sombre apparaîtra plus claire que le fond dans l'image transformée).

Du fait de leur caractère ponctuel, les méthodes de transformation d'histogrammes n'affectent pas la forme des régions. Elles en modifient uniquement l'apparence visuelle. [**Cocquerez & Philipp, 98**]

I.5.1.1. Expansion de dynamique

Cette méthode élémentaire consiste à utiliser au mieux l'échelle de niveaux de gris disponible sur le système d'acquisition d'image dans le but de mieux représenter l'image numérisée.

Soient **A [i, j]** l'image de départ et **A' [i, j]** l'image après transformation.

Soient $[\boldsymbol{a_0}, \boldsymbol{a_1}]$ l'intervalle des intensités présentes dans l'image et $[\boldsymbol{a_{min}}, \boldsymbol{a_{max}}]$ l'intervalle disponible.

L'expansion de dynamique correspond à la transformation linéaire **T** suivante :

$$\boldsymbol{a_s`} = \lambda + \beta\, \boldsymbol{a_{s0}}$$

Telle que :

$$\forall\, a \in [\, a_0, a_1] \xrightarrow{T} \grave{a} \in [\, a_{min}, a_{max}]$$

Avec :

$$\lambda = \frac{(a_{min} * a_1) - (a_{max} * a_0)}{(a_1 - a_0)}$$

$$\beta = \frac{a_{max} - a_{min}}{a_1 - a_0}$$

Cette transformation ne fait qu'améliorer l'aspect visuel de l'image. [**Cocquerez & Philipp, 98**]

Si une image possède des niveaux de gris entre deux valeurs min et max, on va étendre cette plage en le ramenant à une dynamique comprise entre 0 et 255.On peut en déduire le diagramme suivant :

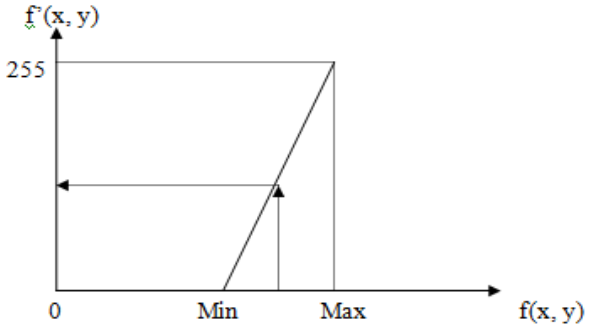


Figure I. 5 : *Schématisation de la transformation par expansion*

I.5.1.2. Égalisation de l'histogramme

L'égalisation d'histogrammes est un outil qui se voit souvent utile pour améliorer certaines images de mauvaise qualité (mauvais contraste, images trop sombres ou trop claires, mauvaise répartition des niveaux d'intensité, etc.…). [**Naciri & Chaoui, 03**]

Cette transformation consiste à rendre le plus plat possible, l'histogramme des niveaux de gris de l'image. On désire que chaque niveau de gris soit équitablement représenté dans l'image.

Soit i l'intensité du pixel de départ, l'intensité de l'image d'arrivée est donnée par f:

$$f(I(x,y)) = 255 * \left(\frac{histocumulé\,(I(x,y))}{nombre\ de\ pixcels\ de\ l'image}\right)$$

Avec : histocumulé (i) : indique le nombre de pixels, (dans l'image I(x, y)) qui ont une intensité inférieure ou égale à i.

I.5.2. Réduction de bruit

Le bruit est toute information ou tout signal qui vient s'ajouter à l'image initiale et qui a pour effet l'altération des caractéristiques de cette dernière. Le traitement du bruit (filtrage) connu aussi sous le nom de réduction de bruit, est une technique importante d'amélioration de la qualité de l'image numérique.

Pour la réduction du bruit, il existe plusieurs méthodes, dont on peut citer :

- Une première méthode consiste à détecter les points incriminés en composant l'intensité de chaque pixel de l'image avec la moyenne des intensités de ces huit voisins. Si l'intensité du pixel s'écarte trop de l'intensité moyenne, on considère que l'on a détecté un parasite, pour l'éliminer on remplace sa valeur par la valeur moyenne. Le point délicat de cette méthode est le choix du seuil nécessaire à la détection des parasites.

- La deuxième méthode est basée sur l'utilisation des filtres linéaire passe-bas. En effet, en lissant l'image, le bruit va diminuer et l'aspect visuel de l'image va être amélioré.

  Les masques les plus utilisés pour le lissage sont :
  - Moyen uniforme : l'intensité du pixel considéré est remplacée par la moyenne des pixels de son voisinage.

$$1/9 * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

  - Moyen accordant un poids important au point central

$$1/10 * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

  - Moyen tenant compte de la distance au point central

$$1/16 * \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

- La troisième méthode est non linéaire, réalisée en appliquant le filtre médian (*détaillé par la suite*). [**Naciri & Chaoui, 03**]

I.5.3. Rehaussement de contraste

Le rehaussement des images est appliqué afin de faciliter l'interprétation visuelle et la compréhension des images. [**Naciri & Chaoui, 03**]

Rehausser le contraste de l'image consiste à diminuer l'étendue de la zone de transition sans affecter l'intensité moyenne des régions situées de part et d'autre (**Erreur ! Source du renvoi introuvable.**). On limite ainsi le risque de fusion intempestive de régions distinctes lors de la phase de segmentation.

L'opérateur de rehaussement de contraste devra, si possible, réduire le bruit dans les zones stationnaires et éviter les phénomènes de dépassement. Ce type d'opérateur est réalisable au moyen de méthodes linéaires ou non linéaires. [**Cocquerez & Philipp, 98**]

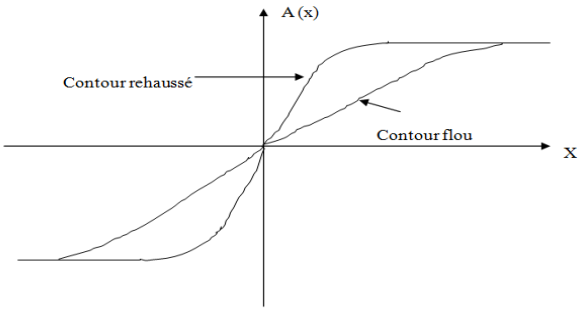


Figure I. 6: *Rehaussement du contraste* [**Cocquerez & Philipp,** 98]

Dans une image brute, les informations utiles sont souvent contenues dans un ensemble restreint de valeurs numériques parmi les valeurs possibles (256 dans le cas de données à 8 bits). Le rehaussement des contrastes se fait en changeant les valeurs initiales de façon à utiliser toutes les valeurs possibles, ce qui permet d'augmenter le contraste entre les cibles et leur environnement.

Il est possible de produire différents types de rehaussements. Il existe plusieurs méthodes qui permettent de rehausser le contraste et les détails d'une image. Nous n'en décrirons que quelques-unes :

- La méthode la plus simple est un rehaussement linéaire du contraste. Afin d'appliquer cette méthode, on identifie les limites supérieures et inférieures d'intensité représentées sur l'histogramme (les valeurs minimales et maximales), et à l'aide d'une transformation linéaire, on étire ces valeurs sur l'ensemble des valeurs disponibles.

  Dans l'exemple suivant, la valeur minimale des données initiales dans l'histogramme est de 84 et la valeur maximale est de 153. Ces 70 niveaux n'occupent qu'un tiers des 256 valeurs disponibles. Un rehaussement linéaire étire de façon uniforme cet intervalle afin d'utiliser la totalité des valeurs de 0 à 255.

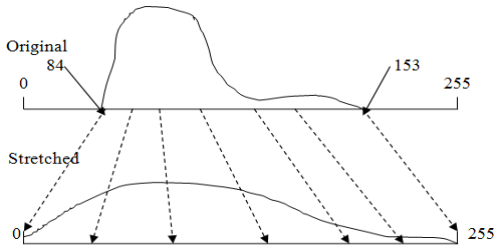


Figure I. 7 : *Rehaussement linéaire du contraste*

  Ce procédé rehausse le contraste dans l'image en pâlissant davantage les régions claires de l'image et en assombrissant davantage les régions plus foncées. Ceci facilite l'interprétation visuelle.

- Un étirement uniforme des valeurs initiales sur l'ensemble des valeurs disponibles n'est pas toujours approprié, particulièrement lorsque la distribution initiale n'est pas uniforme. Dans certains cas, un étirement pondéré de l'histogramme peut donner de meilleurs résultats. Cette méthode assigne un intervalle de valeurs plus grand dans les portions de l'histogramme pour lesquelles les valeurs d'intensité ont une fréquence plus élevée. De cette façon, les détails de ces régions seront mieux rehausser que les détails des régions pour lesquelles la fréquence des valeurs d'intensité de l'histogramme est plus faible. Dans d'autre cas, il peut être préférable de rehausser le contraste dans une portion spécifique de l'histogramme.

- Les filtres spatiaux représentent une autre méthode de traitement numérique dans le cas de rehaussement d'une image. Ces filtres sont conçus de façon à faire ressortir ou supprimer des caractéristiques spécifiques d'une image en se basant sur leur fréquence spatiale.

  La méthode de filtrage spatial consiste à déplacer une fenêtre d'une dimension de quelque pixels (ex : 3 sur 3, 5 sur 5, etc.) au-dessus de chaque pixel de l'image.

  On applique alors un traitement mathématique utilisant les valeurs des pixels sous la fenêtre et on remplace la valeur du pixel centrale par le résultat obtenu. La fenêtre est délacée le long des colonnes et des lignes de l'image, un pixel à la fois, répétant le calcul jusqu'à ce que l'image entière ait été filtrée. En modifiant le calcul effectué à l'intérieur de la fenêtre, il est possible de rehausser ou de supprimer différents types de caractéristiques présentes dans une image. [**Naciri & Chaoui, 03**]

I.6. Techniques de traitement d'images

Dans le processus d'analyse d'une image, la phase de prétraitement a pour but d'améliorer l'homogénéité des régions qui la compose, tout en conservant, voire en renforçant le contraste entre région adjacentes.

Le nombre d'opérateurs, linéaires ou non linéaires, disponibles pour le prétraitement est très important. Le choix d'une méthode, ou de l'enchaînement de traitements élémentaires, dépendra des informations dont on dispose à priori sur le contenu de l'image. Dans le contexte d'application nécessitant des cadences de traitement élevées, il faut aussi tenir compte de l'existence, ou de la possibilité de réalisation, d'opérateurs câblés ou intégrés.

Cependant, les techniques de traitement sont destinées à l'exploitation des informations contenues dans les images, ceci dans le but d'améliorer la qualité des images et de les rendre plus facilement interprétables, en d'autre termes elles permettent d'augmenter la qualité visuelle de l'image, en élevant le rapport signal sur bruit.

Il existe déjà plusieurs techniques qui utilisent l'image sous différents aspects. On utilise son aspect fréquentiel avec les outils de la théorie du signal comme la transformée de Fourrier pour éliminer le bruit. On l'utilise dans son domaine spatial avec les opérateurs de morphologie mathématique pour effectuer des traitements comme l'amincissement, ou la fermeture de contours avec les opérateurs de base que sont la dilatation et l'érosion. On travail aussi dans ce domaine avec des masques qui correspondent à différents filtres et un opérateur appelé la convolution.

I.6.1. Filtrage

Les images brutes permettent rarement de parvenir à une extraction directe des objets à analyser :

- Soit parce que l'éclairement de l'objet n'est pas uniforme;
- Soit parce que l'objet est perçu à travers un bruit assez important : les images contiennent donc un signal et du bruit (dont on veut éliminer la plus grande partie possible);
- Soit encore parce que le contraste n'est pas suffisant. [**Naciri & Chaoui, 03**]

Avant d'extraire les objets et analyser une image, il est donc souvent nécessaire d'améliorer sa qualité. Deux types d'améliorations des images sont possibles :

1. Amélioration du rapport signal sur bruit : La méthode la plus simple pour augmenter le rapport signal/bruit consiste à appliquer le principe des analyseurs multicanaux, c'est-à-dire effectuer plusieurs acquisition de l'image (n par exemple) : ce qui revient à effectuer plusieurs sommations du signal. Le bruit n'apparaissant statistiquement jamais au même endroit, sera uniformément réparti, alors que le signal, apparaissant toujours au même endroit, sera amplifié d'un facteur racine carrée de n.
2. Amélioration par filtrage : L'amélioration du rapport signal/bruit, bien qu'intéressante, est souvent insuffisante pour obtenir de bonnes images. L'amélioration de l'image est essentiellement obtenue par ce que l'on appelle une opération de filtrage.

Le filtrage est un traitement local utilisé principalement pour réaliser une analyse spatiale de l'image, son objectif est d'accentuer les variations d'intensité de l'image, ou de détecter les contours ou de réduire les bruits existants. Il existe un grand nombre de filtres possibles, et à quelques exception prés, on peut les classer en deux grandes catégories: les filtres linéaires et filtres non linéaires.

I.6.1.1. La convolution

Pour mettre en oeuvre un filtrage avec des filtres linéaires, on utilise un opérateur mathématique nommé convolution (noté⊕).

La convolution consiste donc à prendre un voisinage du pixel que l'on veut modifier, le plus souvent ce voisinage est compris dans une fenêtre carrée de côté de longueur impaire, de multiplier chaque valeur du voisinage par des coefficients appartenant à un masque de convolution, d'en faire la somme qui deviendra la nouvelle valeur du pixel. Ce traitement est effectué pour chaque pixel de l'image, mis à part les bords de l'image pour lesquels, soit on met les pixels à 0, soit on modifie le masque et le diviseur. La plus simple approche est d'appliquer l'opération seulement sur une sous image, choisie telle que pour (x, y) dans la sous image, il existe des voisins dans l'image. Ceci donne des résultats ayant un vrai sens, mais on note néanmoins que l'image en sortie produit par l'opération est plus petite que l'image en entrée.

$$\grave{\mathbf{I}}(x,y) = \mathbf{I}(x,y)\,\mathbf{W}(\mathbf{x},\mathbf{y}) = \sum_{m=-d}^{d}\sum_{n=-d}^{d} w(m,n)I(x+m,y+n)$$

Où la taille du masque de convolution est (2d+1)*(2d+1). [**Devaux, 97**]

I.6.1.2. Filtres linéaire

L'utilisation des filtres linéaires provient de l'extension des méthodes mises au point pour le traitement du signal, au traitement des images numériques. Nous pouvons, en effet, considérer une image numérique comme un signal bidimensionnel numérisé.

Un filtre est linéaire s'il peut s'écrire de la manière suivante :

Soit F le filtre utilisé, I1 et I2 représentant deux images alors :

F * (I1+I2) =F*I1 +F*I2 L'opération * représente le produit de convolution.

Les filtres linéaires les plus connus sont les filtres passe-bas, passe-haut. Ces filtres sont représentés sous forme de masque auquel est associé un coefficient diviseur qui permet de ramener les valeurs calculées entre 0 et 255 pour une image à 256 niveaux de gris.

6.1.2.1. Filtre passe-bas

L'intensité du pixel considéré est remplacée par la moyenne des pixels de son voisinage. La taille de la zone (fenêtre) entourant le pixel est un paramètre important, plus cette dimension est grande, plus sa sensibilité au bruit diminue, et le lissage devient important (le flou s'accentue).

Le filtre passe-bas laisse passer les basses fréquences (les faibles changements d'intensité de l'image) et atténue les hautes fréquences (variations rapides).

<u>Exemple de filtre passe-bas</u> :

1/9 *

1	1	1
1	1	1
1	1	1

Masque du filtre

6.1.2.2. Filtre gaussien

L'expression gaussienne en deux dimensions est donnée par : $G_0(x,y) = \frac{1}{2\,\pi\,\sigma^2} * \exp(-\frac{x^2+y^2}{2\sigma^2})$

L'intérêt de ce filtre est que l'on contrôle facilement le degré de filtrage à travers le paramètre **σ**. Le filtre gaussien donne plus d'importance aux pixels proches du pixel central, et diminue cette importance au fur et à mesure que l'on s'éloigne de celui-ci, il a les mêmes inconvénients que le filtre passe-bas, c'est-à-dire il dégrade les contours. La discrétisation de ce filtre pour un σ égale à 0.6 donne le masque suivant :

1/16 *

1	**2**	**1**
2	**4**	**2**
1	**2**	**1**

Masque du filtre

6.1.2.3. Filtre passe-haut

Il est utilisé pour amplifier les détails de hautes fréquences. Il peut permettre par exemple de restaurer des images qui ont été défocalisées et d'accentuer les contours en faisant ressortir les pixels compris entre des zones homogènes. [**Devaux, 97**]

Il met en évidence les changements rapides de l'intensité de l'image (les hautes fréquences) et laisse les zones uniformes inchangées (basses fréquences).

Exemple de filtre passe-haut :

H=

-1	**-1**	**-1**
-1	**9**	**-1**
-1	**-1**	**-1**

Masque du filtre

***I.6.1.3.* Filtres non linéaire**

Ils sont conçus pour régler les problèmes des filtres linéaires, surtout pour ce qui concerne la mauvaise conservation des contours. Leur principe est le même que celui des filtres linéaires, il s'agit toujours de remplacer la valeur de chaque pixel par la valeur d'une fonction calculée dans son voisinage. La différence majeure, est que cette fonction n'est plus linéaire mais une fonction quelconque (elle peut inclure des opérateurs de comparaisons ou de classification). Les filtres non linéaires les plus connu sont :

- Filtre médian Filtre maximum Filtre minimum

6.1.3.1. Filtre médian

Ce filtre est très utilisé pour éliminer le bruit sur une image qui peut être de différentes origines (poussières, petits nuages, baisse momentanée de l'intensité électrique sur les capteurs, ...) et qui se traduit par des taches de faible dimension dont la distribution sur l'image est aléatoire. L'avantage de ce filtre réside dans le fait qu'il conserve les contours alors que les autres types de filtres ont tendance à les adoucir. [**Devaux, 97**] Sur un voisinage à huit, le nouveau niveau de gris du pixel central est choisi comme étant la valeur médiane de tous les pixels de la fenêtre d'analyse, centrée sur ce dernier. Son avantage est qu'il garde la netteté des éléments qui constituent l'image sans étaler les transitions.

L'algorithme de filtre médian est le suivant :

1. Trier les valeurs par ordre croissant.
2. Remplacer la valeur du pixel centrale par la valeur située au milieu de la triée.
3. Répéter cette opération pour tous les pixels de l'image.

Rapport-Gratuit.com

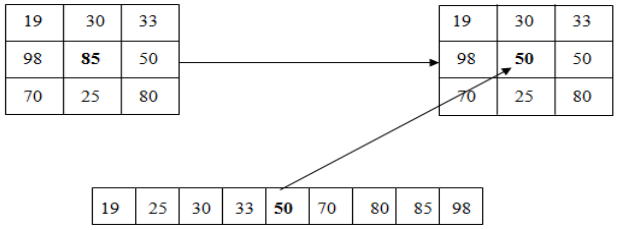


Figure I. 8 : *Principe du filtre médian.*

Exemple :

Image Initiale

Image filtrée

Figure I. 9 : *Application du filtre médian.*

6.1.3.2. Filtre maximum

On applique le même traitement que celui du filtre médian mais la valeur du pixel du centre comme la montre la figure I.14, va être changée par le maximum.

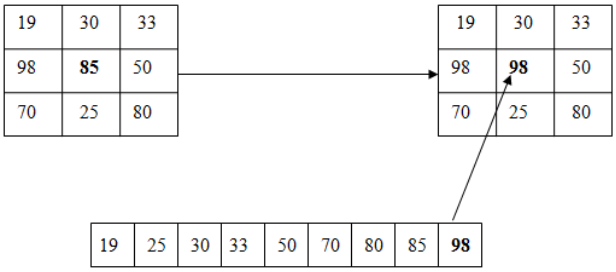


Figure I. 10 : *Principe du filtre maximum.*

Exemple :

Image Initiale

Image filtrée

Figure I. 11 : *Application du filtre maximum.*

6.1.3.3. Filtre minimum

On applique le même traitement que celui du filtre maximum mais, cette fois, la valeur du pixel du centre comme la montre la figure va être remplacée par le minimum.

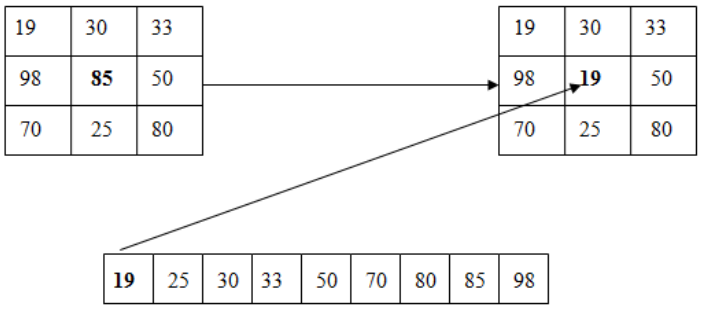


Figure I. 12 : *Principe du filtre minimum.*

I.6.2. Segmentation

Les images sont composées de régions possédant des propriétés locales qui peuvent être la répartition des niveaux de gris. En groupant des points qui possèdent une même propriété donnée, on obtient des régions uniformes. Cette opération s'appelle « segmentation ».

La segmentation est une branche du traitement d'image qui s'occupe de l'analyse d'image ou l'analyse de la scène. Généralement, cette partie reçoit en entrée une image et délivre en sortie une description de cette image (en termes de contours ou de région). La section suivante est consacrée à l'étude des techniques de segmentation d'image.

La segmentation consiste à subdiviser une scène réelle en ses parties constituantes (les objets qui composent l'image), en d'autres termes, elle cherche à extraire les primitives de surface ou de contour. Notons que les pixels d'une primitive sont connexes et homogènes (possèdent la même propriété). [**Besançon, 98**]

Il existe plusieurs méthodes pour la segmentation d'images :

- Classification des pixels.
- Segmentation par détection de contour.
- Segmentation par les régions.

I.6.2.1. Classification des pixels

Ce type de méthode prend en considération uniquement les valeurs de niveaux de gris d'un point pour décider son appartenance à un objet ou pas.

Pour pouvoir le faire, nous devons trouver le domaine de niveaux de gris qui caractérisent l'objet à segmenter. Pour cela, une analyse de l'image et de son histogramme est nécessaire avant de faire le choix en ce qui concerne l'intervalle. Puis, une simple opération de seuillage utilisant les bornes de l'intervalle comme seuils est à effectuer. [**Naciri & Chaoui, 03**]

I.6.2.2. Segmentation par détection de contour

Elle a pour but la détection de discontinuités présentes dans une image, c'est-à-dire les frontières qui existent entre les régions homogènes. Pour cela on effectue des dérivations, car les contours se coïncident avec les points forts du gradient (première dérivée) ou nuls de la dérivée seconde.

Nous appellerons :

- Points de contour : tous points appartenant à la frontière d'une région ;

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

- Contour d'une région : la courbe fermée telle que tous les pixels de la région se trouvent dans un même coté de cette courbe.

Nous citons, dans ce qui suit, quelques opérations sur les contours :

6.2.2.1. Réduction des contours

Elle agit de manière que les contours mis en évidences ne fassent qu'un seul pixel d'épaisseur. Lorsqu'on effectué le gradient d'une image, il est nécessaire d'isoler les maxima locaux de l'image dérivée pour déterminer les points exactes de contours afin de réduire ceci à une courbe d'un seul pixel d'épaisseur.

Mathématiquement, une image est représentée par une fonction f :

$$\boldsymbol{f}: \ \mathbf{IE\,R^2} \rightarrow \mathbf{IF} \subset \mathbf{R}$$

$$\mathbf{P(x,y)} \rightarrow \boldsymbol{f}\mathbf{(x,y)}$$

Donc le gradient de la fonction f est donné par :

$$\text{Amplitude} = \sqrt{\left(\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right)} \qquad \text{Direction} = \left(\left(\frac{\partial f}{\partial x}\right) \ / \left(\frac{\partial f}{\partial y}\right)\right)$$

Dans le cas discret (image digitalisée), ces dérivées sont remplacées par des différences. Par ailleurs on approxime l'amplitude en utilisant une somme ou le maximum des valeurs absolues, plutôt que la racine carrée préconisée. [**Naciri & Chaoui, 03**]

6.2.2.2. Opérateur de ROBERT

Cet opérateur est très sensible aux micros contours, mais aussi aux bruits ; il donne des contours finis mais incomplets, il est donc appliqué aux régions à structure très fine. La qualité des contours est assez bonne. En effet, l'opérateur de ROBERT n'est qu'une estimation des dérivées premières par les expressions Δ_1 et $\Delta_2$données avec la norme et l'orientation :

$$|\nabla f(x,y)| = R(x,y) \qquad |\nabla f(x,y)| = \sqrt{(\Delta_1^2 + \Delta_2^2)}$$

Ceci revient à convoler l'image avec les deux masques suivant :

1	0
0	-1

H1

0	1
-1	0

H2

Exemple :

Image Initiale

Image du contour

Figure I. 13 : *Opérateur de ROBERT.*

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

6.2.2.3. Opérateur de PREWITT

Contrairement à l'opérateur de ROBERT, cet opérateur présente l'avantage d'être moins sensible aux bruits ; il favorise les contours à structure linéaire mais la qualité des contours est meilleure.

Les dérivées directionnelles horizontale et verticale s'expriment sous la forme :

$$\boldsymbol{A_j[i,j] = h_j * A[i,j]} \qquad \text{et} \qquad \boldsymbol{A_i[i,j] = h_i * A[i,j]}$$

Avec

$$\boldsymbol{h_j = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}} \qquad \text{et} \qquad \boldsymbol{h_i = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}}$$

Les matrices $\boldsymbol{h_j}$ et $\boldsymbol{h_i}$, appelées aussi masques, sont les noyaux de convolution de filtres à réponse impulsionnelle finie.

Donc la norme et l'orientation du gradient sont :

$$\boldsymbol{|\nabla A[i,j]| = \sqrt{\left(A_j^2[i,j] + A_i^2[i,j]\right)}} \qquad \text{et} \qquad \boldsymbol{\theta = \frac{\pi}{2} - arctan\left(\frac{A_i}{A_j}\right)}$$

<u>Exemple</u> :

Image Initiale

Image du contour

Figure I. 14 : *Opérateur de PREWITT.*

6.2.2.4. Opérateur de SOBEL

Très sensible aux bruits, il est utilisé pour la détection de contours. Les contours correspondent en général à des changements brusques de propriétés physiques ou géométriques de la scène. Ce filtre comporte deux masques de convolution, un pour la détection des contours verticaux et un autre pour la détection des contours horizontaux. On obtient après convolution une image des gradients. Cette image pourra par la suite être modifiée dans une phase de seuillage pour ne contenir que des pixels à 1 ou 0 correspondant respectivement à un point de contour ou à un non point de contour.

Les dérivées directionnelles horizontale et verticale s'expriment sous la forme :

$$\boldsymbol{A_j[i,j] = h_j * A[i,j]} \qquad \text{et} \qquad \boldsymbol{A_i[i,j] = h_i * A[i,j]}$$

Avec

$$\boldsymbol{h_j = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}} \qquad \text{et} \qquad \boldsymbol{h_i = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}}$$

<u>Exemple</u> :

Image Initiale

Image du contour

Figure I. 6: *Opérateur de SOBEL.*

6.2.2.5. Opérateur du LAPLACIEN

Il approxime de façon très simple la dérivée seconde de l'image. Il est souvent utilisé pour déterminer dans quel coté se trouve un pixel au voisinage d'un bord. En effet, la dérivée seconde est nulle sur ce bord, et prend des signes opposés d'autre part.

C'est un opérateur différentiel à coefficients constants, invariants par rotation. Cet opérateur estime que les points de contours représentent les points où la deuxième dérivée est égale à zéro.

Donc l'opérateur LAPLACIEN:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Peut être exprimé seulement par le masque suivant :

$$L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Et la détection se fait par le produit de convolution entre ce masque et l'image originale.

<u>Exemple</u> :

Image Initiale

Image du contour

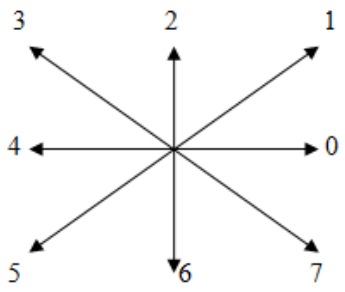
Figure I. 7 : *Opérateur de LAPLACIEN.*

6.2.2.6. Opérateur de KIRSH

L'opérateur de KIRSH est un opérateur à huit masques correspondant chacun à une direction préférentielle et obtenu par rotation de $\frac{\pi}{4}$ de l'opérateur de base h_0 (la numérotation des masques est faite dans l'ordre des directions de Freeman en 8 connexités) :

$$h_0 = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \quad \text{et} \quad h_1 = \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

Avec le codage de Freeman suivant :



Le gradient retenu est donné par :

$$\boldsymbol{max}_{\boldsymbol{i}}\{|\boldsymbol{h_i} * \boldsymbol{A}|; \boldsymbol{i} = \boldsymbol{0}, .., \boldsymbol{7}\}$$

Et l'orientation sera donnée par : [**Cocquerez & Philipp, 98**]

$$\frac{\pi}{\boldsymbol{4}} * \boldsymbol{arg}\ \boldsymbol{max}_{\boldsymbol{i}}\{|\boldsymbol{h_i} * \boldsymbol{A}|; \boldsymbol{i} = \boldsymbol{0}, .., \boldsymbol{7}\}$$

Exemple :

Image Initiale

Image du contour

Figure I. 8 : *Opérateur de KIRSH.*

I.6.2.3. Segmentation par région

Il est très difficile de mettre au point un algorithme de segmentation qui fonctionne correctement dans toutes les situations comme le fait aussi bien le système visuel humain. Nous allons dans ce qui suit exposer quelques techniques de détection des zones homogènes de l'image.

6.2.3.1. Seuillage

Il s'agit d'utiliser l'histogramme de l'image dans le but de partitionner l'image en une ou plusieurs partitions (objets). Chaque classe correspond à une gamme distincte de niveaux de gris. Ceci est réalisé en associant à chaque pic de l'histogramme une classe, le cas ou la segmentation donne seulement deux classes est appelé binarisation.

La binarisation transforme l'image d'entrée en une image binaire c'est-à-dire que les pixels sont soit noir soit blanc. La difficulté majeure de cette transformation est de trouver le bon seuil.

Exemple :

Image Initiale

Image seuillé

Figure I. 9 : Seuillage d'une image.

6.2.3.2. Segmentation par agrégation de pixels

Au départ, la méthode considère chaque pixel comme étant une région, et de regrouper ces régions ensuite selon un double critère: la similarité de niveaux de gris et la connexité. Le critère de similarité peut être par exemple: la variance des niveaux de gris de la région R (et qui doit être inférieure à un certain seuil). L'idée de cet algorithme est de faire croître2 un pixel tant que les éléments de son voisinage vérifient le critère de similarité, sinon on passe à un nouveau pixel et on itère le processus.

6.2.3.3. Segmentation par division

Contrairement à la précédente, celle-ci suppose au départ que tous les pixels forment la même région. Si la région n'est pas homogène (au sens du critère de similarité) on la divise en plusieurs sous-régions (généralement quatre) et on ré-applique l'algorithme, sinon le processus se termine. On peut rajouter à ceci un critère d'arrêt sur la taille des régions.

6.2.3.4. Segmentation par division-fusion (Split & merge)

Cette méthode passe par deux étapes :

- Split : Elle divise l'image en régions (la méthode précédente) et constitue enfin un quad-tree et duquel on va déduire un graphe d'adjacence entre les régions (qui sont les feuilles de l'arbre).
- Merge : Elle rassemble les feuilles du quad-tree qui sont voisines (dans le graphe d'adjacence) et similaires (respectant le critère d'homogénéité) pour avoir de nouvelles régions.

I.6.3. Binarisation

Les traitements les plus connus sur les images binaires sont ceux de la morphologie mathématique, mais leur application sur les images en niveaux de gris est également possible.

Pour avoir des images binaires à partir d'une image grise, on applique les techniques de seuillage (les valeurs supérieures au seuil seront mises à un (1), et les valeurs inférieures au seuil, à zéro (0)).

I.6.3.1. 4.2) Avantage de la binarisation

La binarisation d'une image présente plusieurs avantages dont l'un des plus importants est certainement le faible espace mémoire nécessaire pour cette image binaire et la simplicité des opérations qui lui sont associés. Elle peut être aussi la première étape d'isolement des objets par rapport au fond. De plus, la binarisation constitue une première méthode de la segmentation de l'image.

I.6.3.2. Morphologie mathématique

La morphologie mathématique est connue pour ses relations ensemblistes « $S \subset X$ », « $S \cap X \neq \emptyset$ » où X est l'ensemble à analyser et S un élément structurant que l'on choisit en fonction des besoins de l'analyse.

Ces relations sont, en effet, à la base des opérateurs morphologiques élémentaires que sont l'érosion et la dilatation.

On définit par la suite les mots clés de ce domaine :

Objet : $X = \{x \,/\, x = 1\}$ les pixels blancs de l'image & Fond : $\sim X = \{x \,/\, x = 0\}$ les pixels noirs de l'image

[2] - Croître un pixel = étendre sa région en incluant son voisinage analysé.

Elément structurant B : il peut être assimilé à une fenêtre ou un disque, il est caractérisé par sa forme et sa dimension, il sert à définir un voisinage. [1]

I.6.3.3. Opérations sur les images binaires

Il existe des méthodes qui ont pour objectif l'amélioration des images binaires, parmi celles-ci nous pouvons citer :

6.3.3.1. Dilatation

Elle consiste à dilater l'image, de ce fait les points noirs isolés au milieu des parties blanches sont "mangés" par la dilatation des parties blanches. Ceci veut dire, en d'autres termes, que la dilatation sert à éliminer les points noirs de l'image. Le principe de la dilatation est de parcourir l'image avec une fenêtre carrée de taille N×N (N étant un nombre aléatoire de pixels), et d'effectuer pour chaque pixel de l'image le OU logique des (N-1) voisins, c'est-à-dire tous les pixels de la fenêtre sauf le pixel centrale. Le résultat de cette fonction logique sera :

Si le OU des voisins vaut 1, le pixel courant est forcé à 1 dans l'image résultat.

Si le OU des voisins vaut 0, le pixel courant est simplement recopie de l'image initiale vers l'image résultat. [**Naciri & Chaoui, 03**]

6.3.3.2. Erosion

Elle permet d'éliminer les points blancs de l'image ; le principe est dual à la dilatation, cette fois-ci, on effectue le ET logique de (N-1) pixels voisins ; comme résultat à cette application, l'image est érodée et tout pixel blanc isolé tend à disparaître.

6.3.3.3. Opérations en cascade

À partir des deux transformations de base, on peut définir deux nouvelles opérations qui sont l'ouverture et la fermeture.

- **Ouverture** : c'est une érosion suivie d'une dilatation, elle permet de supprimer certains bruits qui existent dans le fond d'image, et d'isoler certaines surfaces de l'objet.
- ***Fermeture*** : c'est l'opération inverse de l'ouverture, c'est-à-dire que l'on applique d'abord la dilatation ensuite une érosion, elle permet de recoller certains morceaux de surfaces de l'objet et de supprimer les trous à l'intérieur de l'objet.
- ***Squelettisation*** : Le squelette de l'image représente en fait la quantité d'informations minimales qui décrit complètement l'image. Ce qui veut dire qu'il est théoriquement possible de restituer une image originale à partir de son squelette. Donc le squelette est un graphe qui caractérise la forme d'un objet binaire, il représente l'ensemble des points équidistants des deux bords de l'objet. La squelettisation s'effectue par une succession d'opérations appelées amincissement jusqu'à l'obtention d'une structure stable dont les éléments sont les lignes d'épaisseur d'un pixel.

L'amincissement consiste à supprimer de l'objet (mettre à 0) tout pixel ayant une configuration qui coïncide avec l'un des huit éléments structurants (masques) suivants :

1	**1**	**1**
X	**1**	**X**
0	**1**	**0**

Le premier masque de la squelettisation

X : représente 0 ou 1, les autres masques sont déduit par rotation de $\boldsymbol{y} = \frac{\boldsymbol{\pi}}{\mathbf{4}}$ à partir du premier masque (antihoraire). [**Hadjila & Bouabdellah, 03**]

I.7. Conclusion

Les techniques de traitement d'image sont des techniques très diverses et le choix de l'une parmi elles, est un choix qui dépend essentiellement de la nature de l'application et des résultats qui peuvent être obtenus par l'application de l'une ou de l'autre. Cependant, chaque ensemble de ces techniques est destiné à une application spécifique. Certains ensembles ont des applications communes, mais les résultats obtenus seront différents du point de vue avantages et inconvénients.

Notons que la majorité de ces techniques sont supposées comme des opérations de bas niveau, c'est-à-dire qu'elles ne cherchent pas à comprendre la sémantique de l'image, et se limite à la transformation et à la manipulation brute des pixels sans prendre en compte leur signification.

Chapitre II Les Réseaux de neurones

II.1. Introduction

Le rêve de créer une machine dotée d'une forme d'intelligence est présenté depuis longtemps dans l'imagination humaine. Aujourd'hui en intelligence artificielle, on en est encore qu'aux débuts. On arrive tout juste à créer des comportements intelligents primitifs à l'aide des réseaux de neurones qui se sont inspirés des réseaux de neurones biologiques.

Les réseaux de neurones sont en passe de devenir une composante incontournable de l'aide à la décision. Technique fortement stimulée par la vogue du Data Mining, l'analyse neuronale arrive aujourd'hui à un degré de maturité lui permettant de traiter un bon nombre de problèmes.

Les progrès théoriques, l'expérience croissante et l'étude de nouvelles structures neuro-mimétiques semblent très prometteurs pour la résolution quasi-optimale de nombreux problèmes de grande complexité telles que la reconnaissance de formes, le traitement du signal, l'apprentissage, la mémorisation ou la généralisation. Ces applications sont pourtant, malgré tous les efforts déployés en algorithmique et en intelligence artificielle, que très mal imitées par les machines.

Dans ce qui suit, nous allons décrit l'approche neuronal en détaille (définition, les différents architectures, domaine d'application…).

II.2. Historique[3] & définition

II.2.1. Historique

Le champ des réseaux neuronaux va démarrer par la présentation en *1943* 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). En *1949*, D. Hebb présente la première règle d'apprentissage. De nombreux modèles de réseaux aujourd'hui s'inspirent encore de la règle de Hebb. En *1958*, F. Rosenblatt développe le modèle du perceptron. C'est le premier système artificiel capable d'apprendre par expérience. Dans la même période, Le modèle de L'Adaline (ADAptive LINar Element) a été présenté par B. Widrow et Hoff. 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. Cela va avoir une grande incidence sur la recherche dans ce domaine. Elle va fortement diminuer jusqu'en *1972*, où T. Kohonen présente ses travaux sur les mémoires associatives et propose des applications à la reconnaissance de formes. En *1982*, Hopfield a remis à la mode le connexionnisme en donnant une architecture solide inspirée de la physique.

II.2.2. Définition

Les réseaux de neurones informatiques sont une partie de l'intelligence artificielle dont l'application s'élargit de jour en jour. Parmi les plus connues, la lecture des codes postaux, la reconnaissance de

[3] Cours Master « Introduction aux réseaux de neurones », année 2004/2005 (dernière consultation Septembre 2011), http:// www.univ-bpclermont.fr/atmos/enseignement/cours_RN_2006.pdf

Rapport-Gratuit.com

visages, le filtrage d'images, la reconnaissance des formes. Cette technique s'utilise quand certaines conditions sont réunies en voici deux principales :

Pas de règles clairement définies beaucoup d'exemples

Le réseau se compare un peu à une boite noire : on met d'un côté les informations et de l'autre on obtient une décision. Dans un réseau informatique les informations sont en général présentées comme allant de gauche à droite. Chaque neurone collecte les informations venant des neurones précédents et en déduit une information à transmettre aux neurones suivants. Chaque neurone donne plus ou moins d'importance à certaines des informations qu'il reçoit. L'apprentissage consiste à trouver les ajustements pour qu'il accorde la bonne importance à la bonne information. Pour cela, à chaque prévision faite par le réseau au moment de son apprentissage on lui présente la réponse qu'il aurait dû sortir. Il analyse les écarts pour s'adapter. Cet apprentissage supervisé a pour inconvénient que l'on ne saura pas comment fait le réseau pour travailler. Par contre, une fois l'apprentissage fait, en utilisation le réseau ne fait que des opérations mathématiques, ce qui permet de faire un travail à une rapidité permettant des applications en temps réel. [14]

II.3. Neurone biologique

C'est l'organe élémentaire de base qui forme le système nerveux des être vivant, il servi à la mise au point des premiers neurones formels.

Cette partie permet de présenter rapidement les éléments de base du cerveau que sont les neurones et d'avoir un bref aperçu sur leur fonctionnement.

Le neurone biologique est composé des éléments suivants : le corps cellulaire, l'axone, les dendrites et les synapses

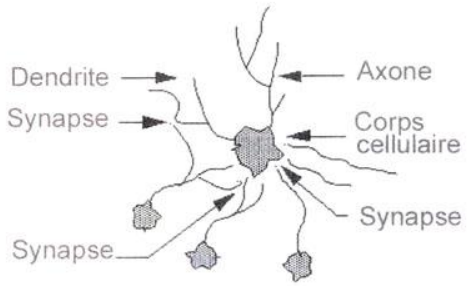


Figure II. 10 : *le neurone biologique.*

II.3.1. Le neurone

Les neurones sont les cellules nerveuses à la base du système nerveux central. Chez l'humain, on en compte environ cent milliards.

Le neurone est constitué d'un corps cellulaire et d'un noyau. Il se ramifie de dendrites par où s'achemine l'information vers le neurone. Une fois l'information traitée par le neurone elle est envoyée sur l'axone. Et la jonction entre deux neurones s'appelle synapse. Les informations parvenant au neurone peuvent venir d'autres neurones ou de capteurs sensoriels tels que le touché, la vue, l'odorat…etc. ou d'autres capteurs internes à l'organisme. Le corps cellulaire contient le noyau du neurone et effectue les transformations biochimiques nécessaires à la vie du neurone. Les dendrites sont de fins tubes de quelques dixièmes de micron de diamètre et d'une longueur de quelques dizaines de microns. Ceux sont des petites branches qui se ramifient à leur extrémité. Elles forment une sorte d'arborescence autour du corps cellulaire. Ceux sont elles qui permettent au neurone de capter les

signaux qui parviennent de l'extérieur. L'axone est la fibre nerveuse qui permet de transporter les signaux émis par le neurone. Sa membrane externe possède certaines propriétés, il est plus long que les dendrites (un millimètre à plus d'un mètre) et se ramifie à son extrémité là où il communique avec les autres neurones.

Deux neurones sont connectés entre eux dans des endroits qui s'appellent les synapses. Ils ne sont pas directement reliés l'un à l'autre, mais sont séparés par un petit espace, appelé espace synaptique. Les synapses jouent un rôle fondamental pour permettre aux neurones de communiquer. Certains traitements sont effectués à leurs niveaux. [12]

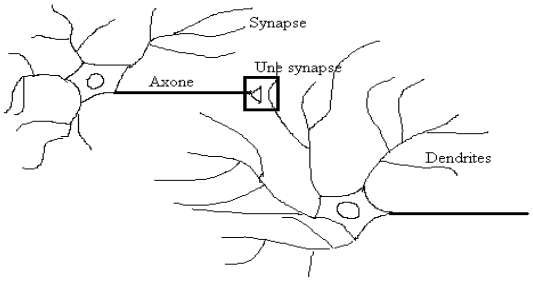


Figure II. 11 : *Connexion entre les neurones.*

II.3.2. Le fonctionnement

Le fonctionnement classique d'un neurone, consiste en générale à recevoir et émettre des signaux électriques, est cela comme suit :

Au niveau de chaque neurone on a une intégration des signaux reçus (qui parviennent des autres neurones) au cours du temps, c'est en quelque sort une sommation des signaux reçus qui se produit au niveau de chaque neurone, quand cette somme dépasse un certain seuil, ce dernier émet à son tour un signal électrique. Ainsi les neurones communiquent en émettant des trains de potentiels rapides et très courts de l'ordre de quelques millisecondes.

En générale, la membrane externe d'un neurone exécute cinq fonctions :

- Propagation des impulsions électriques le long de l'axone et des dendrites
- Libération des médiateurs à l'extrémité de l'axone
- Réagir à ces médiateurs chimiques au niveau des dendrites
- Réagir au niveau du corps cellulaire aux impulsions électriques issues des dendrites pour générer ou non une nouvelle impulsion
- Permettre au neurone de reconnaître ses voisins pour se situer au cours de la formation du cerveau et trouver à quelles autres cellules il doit ce connecté.

II.4. Neurone formel

Un neurone formel est en général une fonction algébrique non linéaire et bornée, dont la valeur dépend de paramètres appelés coefficients ou poids. Les variables de cette fonction sont habituellement appelées « entrées » du neurone, et la valeur de la fonction est appelée sa « sortie ».

II.4.1. Synoptique

C'est un modèle mathématique très simple dérivé d'une analyse de la réalité des neurones biologiques. Pour le réaliser, on effectue une simple sommation des signaux arrivant au neurone, la sortie est obtenue par comparaison de cette somme au seuil de déclanchement, par exemple, elle peut être égale à 1 si la somme est supérieure au seuil et 0 dans le cas contraire.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Plus simple, il suffit de soustraire le seuil considéré à la somme des entrées, et de faire passer le résultat par la fonction de transfère, la sortie du neurone est le résultat obtenu après le transfère, ainsi, on obtient une représentation des propriétés physique du neurone biologique. Pour donner plus de puissance à ce modèle, on introduit la connexion synaptique qui est une valeur réelle, la sortie d'un neurone est alors multipliée par la valeur de la connexion synaptique durant le traitement effectué par le neurone récepteur.

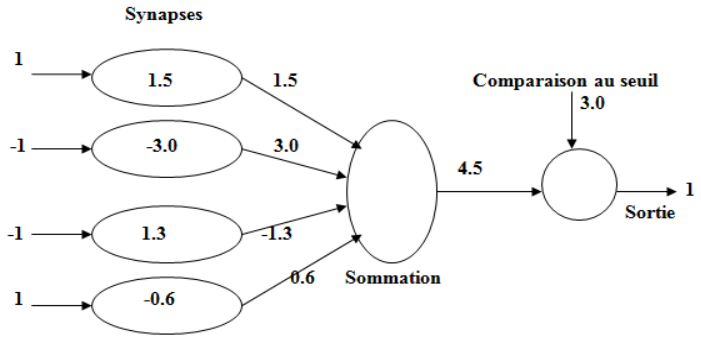


Figure II. 12 : *fonctionnement du neurone formel*

En résumer, on peut dire que le neurone formel réalise tout simplement une somme pondérée de ces entrées, ajoute un seuil à cette somme et fait passer le résultat par une fonction de transfert pour obtenir sa sortie. [**El Hannani, Koudacke, 2002**]

II.4.2. Le fonctionnement

Pour fonctionner, un neurone formel utilise des entrées qui sont des grandeurs réelles. Si on relie un neurone au monde extérieur par des capteurs, il peut réaliser une simple analyse de ce qu'il « perçoit ». Si on représente les valeurs observées par le neurone sous forme d'un vecteur, le neurone réalise alors un découpage de son espace d'entrée (l'espace vectoriel auquel appartient le vecteur d'observation) en deux zones :

- La zone d'activité : dont les vecteurs donnent une sortie égale à 1 ;
- La zone d'inactivité : dont les vecteurs donnent une sortie égale à 0.

Comme le calcul effectué est en fait linéaire, la séparation l'est aussi. Les coefficients synaptiques et le seuil définissent l'équation d'un hyperplan qui est la frontière de la séparation entre les deux zones.

Mathématiquement, le neurone formel est considéré comme une fonction non linéaire, bornée et dont la valeur de sortie dépend de certains paramètres appelés coefficients ou poids. Les entrées de cette fonction représentent les entrées du neurone, tandis que son résultat représente la sortie. [**Jadouin, 94**]

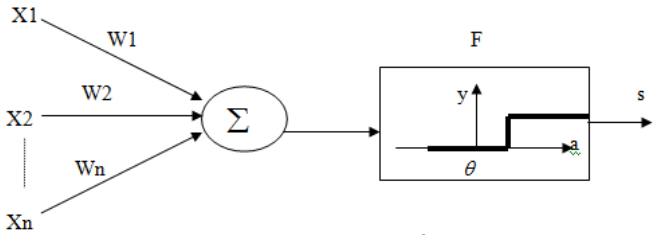


Figure II. 13 : *Neurone formel.*

Les Xj sont les n signaux d'entrée, et les Wij représentent les poids qui relient les neurones i, j; L'activation pondérée « a » est donnée par :

$$a = x_1 w_{i1} + x_2 w_{i2} + \cdots + x_n w_{in} = \sum x_j w_{ij} \qquad y = F(a)$$

La sortie finale est donnée par le seuil d'activation comme suit : $y = \begin{cases} 1, & si\ a \geq 0 \\ 0, & si\ a < 0 \end{cases}$

La fonction d'activation f, détermine l'état interne du neurone en fonction de son entrée totale. Plusieurs modèles de cette fonction existent dans la littérature, on les décrit par la suite.

Fonction Heaviside définie par : $H(x) = \begin{cases} 1, & si\ x \geq 0 \\ 0, & sinon \end{cases}$

Fonction signe définie par : $Signe(x) = \begin{cases} +1, & si\ x \geq 0 \\ -1, & sinon \end{cases}$

La fonction linéaire est définie par : $F(x) = x$

La fonction sigmoïde est définit par : $F(x) = \dfrac{1}{1 + e^{-x}}$

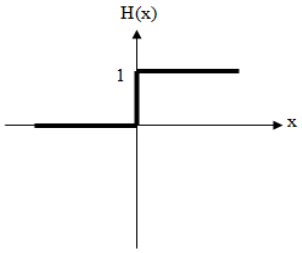


Figure II. 14 : *fonction Heaviside.*

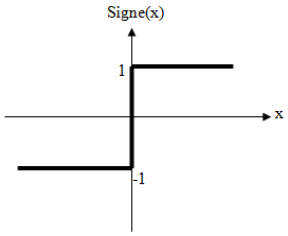


Figure II. 15 : *fonction signe.*

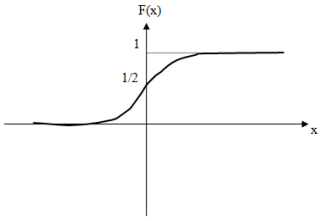


Figure II. 16 : *fonction sigmoïde.*

Donc on peut conclure qu'un neurone formel réalise simplement une somme pondérée de ces entrées, ajoute un seuil à cette somme et fait passer le résultat par une fonction de transfert pour obtenir sa sortie.

II.5. Architecture des réseaux de neurones

Les réseaux neuromimétriques peuvent se répartir en deux grandes catégories : la première est dite « réseaux à couches », la seconde est appelée « réseaux récurrents » qui présente certaine complexité par rapport à la première.

La connectivité des réseaux peut être totale, c'est-à-dire que tous les neurones sont connectés entre eux ou bien, par couches, et dans ce cas les neurones d'une couche ne sont connectés qu'aux neurones de la couche suivante [**Jadouin, 94**].

II.5.1. Réseaux multicouches

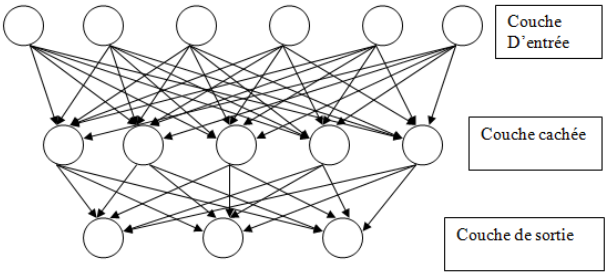


Figure II. 17 : *Réseau de neurone non bouclé.*

Les neurones sont arrangés par couches. Il n'y a pas de connexion entre neurones d'une même couche et les connexions ne se font qu'avec les neurones des couches en avale. Habituellement, chaque neurone d'une couche est connecté à tous les neurones de la couche suivante et celle-ci seulement. Par

extension, on appelle couche d'entrée l'ensemble des neurones d'entrée, couche de sortie l'ensemble des neurones de sortie. Les couches intermédiaires n'ayant aucun contact avec l'extérieur sont appelées couches cachées.

II.5.2. Réseaux à connexions locales

Il s'agit d'une structure multicouche. Chaque neurone entretien des relations avec un nombre réduit et localisé de neurones de la couche en aval. Les connexions sont donc moins nombreuses que dans le cas d'un réseau multicouche classique.

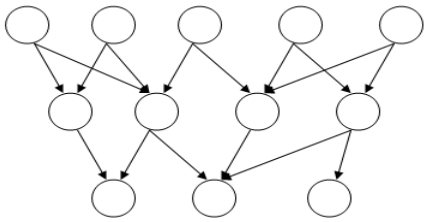


Figure II. 18 : *Réseau à connexions locales.*

II.5.3. Les réseaux de neurones récurrents

Nous trouvons des boucles, soit au niveau des neurones, soit entre les couches, on distingue dans cette catégorie les réseaux à connexion symétriques, ils sont utilisés généralement, dans la modélisation de systèmes dynamiques, la détection automatique des clusters [**Jadouin, 94**]….

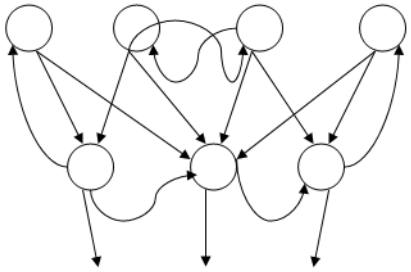


Figure II. 19: *Réseau de neurone bouclé.*

II.5.4. Les réseaux de neurones à connexion complète

C'est la structure d'interconnexion la plus générale. Chaque neurone est connecté à tous les neurones du réseau (et à lui-même). [13]

Il existe de nombreuses autres architectures possibles, mais celles ci sont les plus utilisées.

II.6. Apprentissage des réseaux de neurones

II.6.1. Définition

L'apprentissage est une phase de développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré.

On distingue trois grandes classes d'algorithmes d'apprentissage : apprentissage supervisé, apprentissage non supervisé, et apprentissage par renforcement.

II.6.2. Techniques d'apprentissage

II.6.2.1. Apprentissage supervisé

Dans ce cas on fournit au réseau la donnée à traiter mais aussi la sortie attendu. Les résultats corrects (c'est-à-dire les valeurs que l'on désire que le réseau obtienne en sortie) sont fournis au réseau, si bien que celui-ci peut ajuster ses poids de connexions pour les obtenir (la modification des paramètres

internes afin de minimiser l'erreur constatée). Après l'apprentissage, le réseau est testé en lui donnant seulement les valeurs d'entrée mais pas les sorties désirées, et en regardant si le résultat obtenu est proche du résultat désiré.

II.6.2.2. Apprentissage non supervisé

Pour des réseaux à apprentissage non supervisé, on ne fournit pas au réseau les sorties que l'on désire obtenir. On le laisse évoluer librement jusqu'à ce qu'il se stabilise (s'auto-organise), o, utilisant les lois régissant l'évolution des synapses.

II.6.2.3. Apprentissage par renforcement

L'apprentissage est de type par renforcement lorsque le réseau de neurone interagit avec l'environnement. L'environnement donne récompense pour une réponse satisfaisante du réseau et assigne une pénalité dans le cas contraire. Le réseau doit ainsi, découvrir les réponses qui lui donnent un maximum de récompenses.

II.7. Apprentissage dans quelques réseaux célèbres

II.7.1. Perceptron

Un perceptron linéaire à seuil prend en entrée n valeurs x1,..., xn et calcule une sortie o. Un perceptron est défini par la donnée de n+1 constantes : les coefficients synaptiques w1, ..., wn et le seuil (ou le biais) q. La sortie « *o* » est calculée par la formule :

$$o = \begin{cases} 1 & \sum w_i x_i > \theta \\ 0 & sinon \end{cases}$$

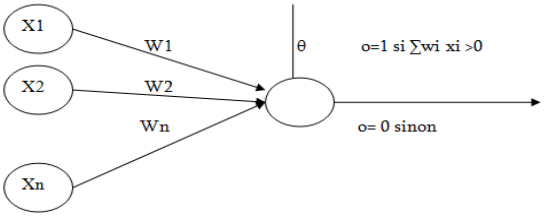


Figure II. 20: *Le perceptron avec seuil*

Les entrées x1,..., xn peuvent être à valeurs dans {0,1} ou réelles, les poids peuvent être entiers ou réels. Une variante très utilisée de ce modèle est de considérer une fonction de sortie prenant ses valeurs dans {-1,1} plutôt que dans {0,1}. Il existe également des modèles pour lesquels le calcul de la sortie est probabiliste. Dans la suite de cette partie sur le perceptron, nous considérerons toujours le modèle déterministe avec une sortie calculée dans {0,1}.

Pour simplifier les notations et certaines preuves, nous allons remplacer le seuil par une entrée supplémentaire x_0 qui prend toujours comme valeur d'entrée la valeur x_0=1. À cette entrée est associé un coefficient synaptique w0. Le modèle correspondant est décrit dans la figure suivante. On peut décomposer le calcul de la sortie « *o* » en un premier calcul de la quantité $\sum w_i x_i$ appelée *potentiel post-synaptique* ou l'*entrée totale* suivi d'une application d'une *fonction d'activation* sur cette entrée totale. La fonction d'activation est la fonction de Heaviside définie par :

$$o = \begin{cases} 1 & si\ x \geq 0 \\ 0 & sinon \end{cases}$$

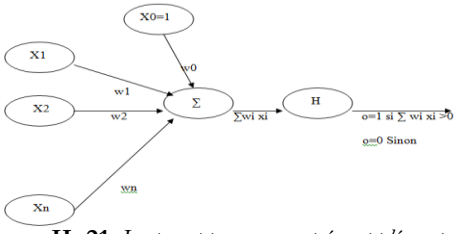


Figure II. 21: *Le perceptron avec entrée supplémentaire*

Pour passer du modèle avec sorties à valeurs dans {0,1} au modèle à valeurs dans {-1,1}, il suffit de remplacer la fonction de Heaviside *f* par la fonction g définie par : *g(x) = 2f(x) - 1*. D'autres fonctions d'activation peuvent également être utilisées.

L'apprentissage du perceptron est de type supervisé. Cela consiste à présenter une série d'exemples en entrées X et la valeur désirée en sortie. La modification des poids s'effectue grâce à la règle suivante :

$$\boldsymbol{w_{ij}(t+1) = w_{ij}(t) + (t_j - o_j)x_i = w_{ij}(t) + \Delta w_{ij}}$$

- o_j représente la sortie d'un neurone
- t_j représente la sortie désirée
- W les valeurs des connections [**Devaux, 97**]

Concernant l'évaluation du, différents problèmes se rapportant au perceptron ont été cités dans la littérature. L'un des principaux rencontrés est que le perceptron ne peut pas résoudre des problèmes de classification non linéaires comme par exemple la fonction logique XOR (ou exclusif).

Plusieurs techniques ont été créées pour essayer de résoudre les problèmes posés par le perceptron, comme le recodage des entrées ou les réseaux à couches cachées.

II.7.2. Réseaux multicouches

Un réseau multicouches possède une structure comme représentée figure 10. Les couches comprises entre les entrées et la couche de sorties correspondent aux couches cachées. Chaque neurone de chaque couche possède une liaison avec tous les neurones de la couche suivant.

Ce qui diffère par rapport au perceptron, c'est la fonction d'activation d'un neurone. Dans le perceptron, un seuil détermine l'état de la sortie, alors que pour un neurone d'un réseau multicouche, on utilise une fonction continue et dérivable, le plus souvent la fonction sigmoïde dont la forme est représentée figure 11 et l'équation est :

$$\boldsymbol{F(x) = 1/(1 - e^{-x})}$$

Où x représente la valeur d'activation d'une cellule.

Cette fonction, de par sa forme, permet d'avoir en entrée et en sortie des valeurs continus comprises entre 0 et 1. [**Devaux, 97**]

<u>Algorithme d'apprentissage</u> :

Avant de pouvoir être utilisé, le réseau doit pouvoir déterminer la valeur des poids de chacune de ses connections. Cette valeur est déterminée durant la phase d'apprentissage on utilisant la règle de rétro propagation du gradient. Cette règle permet de déterminer les valeurs des poids dans toutes les couches du réseau en fonction de l'erreur entre la valeur obtenue et la valeur désirée, donc le but de l'apprentissage est de minimiser cette erreur.

On définit une erreur totale $\mathbf{E_p}$ qui permettra de stopper l'apprentissage. L'équation de $\mathbf{E_p}$ est la suivante :

$$\mathbf{E_p} = \frac{1}{2}\sum_{m=0}^{M}(t_m - o_m)^2$$

Où

M représente le nombre de sortie du réseau

$\mathbf{t_m}$ la sortie désirée

$\mathbf{o_m}$ la sortie calculée

Pour tous les exemples de la base d'apprentissage, on obtient : $\mathbf{E} = \sum_{p=0}^{\mathbf{P}} \mathbf{E_p}$

L'algorithme d'apprentissage est le suivant :

1. Initialiser les poids : On attribut à chaque poids une valeur aléatoire comprise entre -1 et +1.
2. Présenter un exemple au réseau : On présente un exemple Xp en entrée et la valeur désirée en sortie.
3. Propager les signaux d'activation des entrées vers les sorties : On calcul les sorties de la première couche qui deviennent les entrées de la couche suivante, puis on fait de même jusqu'à la dernière couche. Pour un réseau contenant C couches on a :

   $$\mathbf{O_j^c} = \boldsymbol{sigmoid}\left(\sum_{i=0}^{n^{c-1}} \boldsymbol{w_{ij}^c o_i^{c-1}}\right)$$

   Avec $\boldsymbol{w_{ij}^c}$ poids de la connexion entre le neurone i de la couche c-1 et le neurone j de la couche c.
4. Comparer avec les sorties désirées et rétro propager l'erreur des couches de sortie.
   On calcul l'erreur pour chaque neurones avec l'équation suivante :

   Pour les neurones de sortie :

   $$\boldsymbol{e_j = o_j(1 - o_j)(t_j - o_j)}$$

   Pour les neurones en couches cachées :

   $$\boldsymbol{e_j^c = o_j^c(1 - o_j^c) \sum_{k=0}^{n^{c+1}} e_k^{c+1} w_{jk}^{c+1}}$$

5. Modifier les poids
   Pour c de C à 1,
   Pour j de 0 à $\boldsymbol{n^{c-1}}$, faire

   $$\boldsymbol{w_{ij}^c(t+1) = w_{ij}^c(t) + \varepsilon\, e_j^c o_j^{c-1} + \eta \Delta w_{ij}^c(t-1)}$$

   Où $\boldsymbol{\eta}$ représente le taux d'apprentissage.
6. Recommencer.
   On recommence tant que E est supérieur à l'erreur minimale souhaitée.
7. La constante $\boldsymbol{\varepsilon}$ est le momentum term $\boldsymbol{\eta}$. Elle permet de limiter les oscillations lors de l'apprentissage en prenant en compte les évolutions des étapes précédentes dans l'étape courante. Elle permet donc de converger plus rapidement vers la solution recherchée.

<u>Evaluation du réseau multicouche</u> :

Ce type de réseau et l'algorithme de rétro propagation de l'erreur ont fait revivre l'intérêt des chercheurs pour les réseaux de neurones. Un des avantages de ce type de réseau est que l'apprentissage est fait localement, donc une implémentation parallèle est possible.

Les problèmes avec l'algorithme de rétro propagation de l'erreur sont dus au temps de convergence très lent, au fait que durant l'apprentissage, le réseau peut se retrouver bloqué dans un minimum local et que si les poids deviennent trop élevés, leur changement devient minimal. De plus, la réussite de l'apprentissage dépend aussi de la valeur initiale des poids car la méthode du gradient converge toujours vers le minimum le plus proche du point de départ.

II.7.3. ADALINE

L'ADAptive LINear Element a la même architecture que le perceptron, mais en revanche, il possède des valeurs d'activations continues, une fonction d'activation linéaire est une règle d'apprentissage un peu développée que celle du perceptron, elle s'appelle la règle de WIDROW & HOFF, en effet cette loi minimise l'erreur quadrique, en utilisant la méthode du gradient.

Contrairement à la règle du perceptron cette dernière opère sur les exemples bien classés et mal classés en optimisant toujours les coefficients des hyperplans séparateurs.

Formellement, on définit la dynamique ainsi que la règle d'apprentissage comme suit [**Jadouin, 94**] :

$$S_i = \sum (W_{ij} * X_j - \theta_i)$$

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$$

$$\Delta w_{ij}(t) = \alpha * S_j * (d_i - X_i)$$

Avec :

S_i : sortie réelle du neurone j.

α : Le pas d'apprentissage.

X_i : La sortie réelle du neurone i.

d_i: La sortie désirée du neurone i.

θ_i: Le seuil associé au neurone i.

t : Numéro de l'itération.

II.7.4. Les réseaux de HOPFIELD

Développé en 1982, ce réseau est récurrent et se base sur le principe des mémoires associatives, il est constitué d'une couche de neurones totalement connectées (symétriquement $W_{ij} = W_{ji}$).

Un réseau de HOPFIELD réalise une mémoire adressable par son contenu, il s'agit d'un réseau constitué de neurones de MC CULLOCH et PITTS (à deux états, -1 et 1 ou 0 et 1), dont la loi d'apprentissage est la règle de HEBB (1949) qui s'inspire de la neurobiologie, son principe est qu'une liaison synaptique améliore son activité si seulement si l'activité de ses deux neurones est corrélée, c'est-à-dire que les poids W_{ij} d'une connexion reliant un neurone i et un neurone j augmente quand les deux neurones sont activés en même temps, sinon le poids reste inchangé [**Hadjila & Bouabdellah, 03**].

Concernant la dynamique du réseau, si nous considérons un réseau contenant n neurones complètement connectés, à chaque neurone i, on associe un état Vi (c'est-à-dire sa sortie) et on note t_{ik} la connexion entre deux neurones i et k, avec t la matrice des interconnexions (symétrique) et dont les éléments diagonaux W_{ii} sont nuls.

L'activation discrète d'un neurone i est donnée par :

$$V_i = h(u_i) = \begin{cases} 1 & si\ U_i \geq 0 \\ 0 & sinon \end{cases}$$

$$u_i = \left(\sum_{k=1}^{N} t_{ik} * V_k \right) - \theta_i$$

Avec :

h : la fonction Heaviside définie précédemment.

$k \neq i$ et θ_i le seuil associer au neurone i

Pour la fonction d'énergie du réseau : Soit, I= (V1, V2, V3,…, Vn) l'état du réseau constitué par l'ensemble des états élémentaires de chaque neurone. Partant d'un état initial du réseau qui représente le problème, on veut atteindre un état stable qui correspond au résultat du calcul (solution du problème), tout en minimisant une fonction E(i) appelée énergie totale.

$$E(i) = -\frac{1}{2}\sum_{i,j}(W_{ij} * V_i * V_j)$$

Suivant la règle définie plus haut, l'évolution du réseau se fait à énergie décroissante, ainsi les états à énergie décroissante, ainsi les états à énergie minimum sont des points fixes pour l'évolution du réseau. Formellement la règle s'applique comme suit [**Jadouin, 94**]:

$$W_{ij}(t+1) = W_{ij}(t) + \alpha * (V_i * V_j)$$

Avec α : Le pas d'apprentissage.

II.7.5. Les réseaux de KOHONEN

Ces réseaux sont inspirés des observations biologiques du fonctionnement des systèmes nerveux de perception des mammifères. La figure suivante indique que le réseau est constitué de deux couches, la première contient les patrons d'entrées, la deuxième constitue les sorties associées. Chaque neurone de cette dernière est connecté à tous les neurones d'entrées à travers n connexions qui portent les poids w_{ij} .

Ce modèle utilise un apprentissage compétitif, dont le principe est qu'un seul neurone sera activé, après la propagation, ce dernier sera ensuite encouragé (C'est-à-dire que le neurone gagnant tout seul, selon une certaine fonction, ainsi que, dans certains cas, un certain voisinage de ce nœud, verra son poids modifié). On calcule le neurone gagnant4 en choisissant, celui dont le vecteur des poids est le plus proche du vecteur d'entrées selon une distance euclidienne. L'apprentissage consiste à rapprocher les poids des neurones gagnants vers les valeurs d'entrées.

Les cartes de Kohonen se trouvent utiles dans le cas d'une classification automatique. En effet, on utilise les neurones de la couche de Kohonen comme des indicateurs d'appartenance à une classe donnée.

Pour un certain neurone, le vecteur W de poids qui lui est associé indique le représentant de la classe que ce neurone à la tâche de symboliser. Ce neurone ne sera activé que si l'élément présenté au réseau est plus près du représentant de cette classe et que les représentants des autres classes sont plus ou moins loin. Bien sur, le problème de l'utilisation de réseaux de Kohonen pour effectuer ce type de classification automatique est dans le nombre de classes, et en même temps, dans le nombre de neurones de la couche de Kohonen. Notons que le réseau de Kohonen trouvera des classes même s'il n'en existe pas. [**Jadouin, 94**]

Concernant la dynamique, L'activation du réseau de Kohonen par application du vecteur d'entrée X, revient à trouver le neurone gagnant j (recherche de la distance euclidienne minimale) :

$$j(p) = min_i\{\|X - W_i(p)\|\} \ ; i = 1, 2, \dots, n$$

Le i parcourt la couche de sortie et m représente le nombre de neurones de la couche de décision (sortie).

L'apprentissage est réalisé de la manière suivante [**Jadouin, 94**] :

$$\nabla W_i k(p+1) = \begin{cases} \alpha * [Xk - W_i k(p)], & i \in A_j(p) \\ 0, & i \notin A_j(p) \end{cases}$$

Où : $A_j(p)$: le voisinage du neurone gagnant j. P: le numéro de l'itération.

[4] - neurone gagnant = neurone prépondérant, celui qui influence le plus le résultat désiré.

Rapport-Gratuit.com

K : parcourt tout les neurones d'entrée. α : Le pas d'apprentissage.

$\mathbf{W}_i$: Le vecteur des poids associé au neurone de sortie i.

II.7.6. Les réseaux RBF

Les réseaux RBF (Radial Basic Function) ont une architecture en couches, on trouve exactement une couche d'entrée, une couche cachée et une couche de sortie, ils ressemblent fortement aux PMC mais avec des différences importantes :

Chaque neurone caché comporte une fonction noyau (par exemple gaussienne) au lieu de la sigmoïde. Pour un point donné de l'espace d'entrées, la sortie d'un neurone caché est égale à la hauteur de la gaussienne en ce point, elle est plus forte dans le centre du noyau. La sortie du réseau est une combinaison linéaire des sorties des neurones cachés. L'utilisateur doit choisir le nombre de neurones cachés, l'algorithme d'apprentissage définit alors les paramètres des gaussiennes, et calcule les coefficients de la combinaison linéaire de sortie (c'est-à-dire les poids entre la couche cachés et celle de sortie).

Un réseau RBF est plus facile à construire qu'un PMC, et apprend beaucoup plus vite, mais sur les problèmes difficiles les réseaux RBF ne peuvent pas lutter avec les réseaux PMC, en terme de qualité des résultats. [**Jadouin, 94**]

II.8. Les étapes de la conception d'un réseau[5]

Les débutants sont souvent surpris d'apprendre que pour construire un réseaux de neurones, la première chose à faire n'est pas de choisir le type du réseau mais de bien choisir ses échantillons de données d'apprentissage, de test et de validation. C'est n'est qu'ensuite que le choix du type du réseau interviendra. Afin de clarifier les idées, nous présentons dans ce qui suit les quatre étapes (chronologiquement parlant) qui doivent guider la création d'un réseau de neurone.

II.8.1. Choix des échantillons

Le processus d'élaboration d'un réseau de neurones commence toujours par le choix et la préparation des échantillons de données. Comme dans les cas d'analyse de données, cette étape est cruciale et va aider le concepteur à déterminer le type de réseau le plus approprié pour résoudre sont problème. La façon dont se présente l'échantillon conditionne : le type de réseau, le nombre de cellules d'entrée, le nombre de cellules de sortie et la façon dont il faudra mener l'apprentissage, les tests et la validation.

II.8.2. Elaboration de la structure du réseau

La structure du réseau dépend étroitement du type des échantillons. Il faut d'abord choisir le type de réseau : un perceptron standard, un réseau de Hopfield, un réseau à décalage temporel (TDNN), un réseau de Kohonen, un ARTMAP etc... Dans le cas du perceptron par exemple, il faudra aussi choisir le nombre de neurones dans la couche cachée. Plusieurs méthodes existent et on peut par exemple prendre une moyenne du nombre de neurones d'entrée et de sortie, mais rien de vaut de tester toutes les possibilités et de choisir celle qui offre les meilleurs résultats.

II.8.3. Apprentissage

L'apprentissage consiste tout d'abord à calculer les pondérations optimales des différentes liaisons, en utilisant un échantillon. La méthode la plus utilisée est la rétro propagation : on entre des valeurs du

[5] http://www.sylbarth.com/nn.php

les cellules d'entrée et en fonction de l'erreur obtenue en sortie (le delta), on corrige les poids accordés aux pondérations. C'est un cycle qui est répété jusqu'à ce que la courbe d'erreurs du réseau ne soit croissante (il faut bien prendre garde ne pas sur-entrainer un réseau de neurones qui deviendra alors moins performant). Il existe d'autres méthodes d'apprentissage telles que le quickprop par exemple.

II.8.4. Validation et tests

Alors que les tests concernent la vérification des performances d'un réseau de neurones hors échantillon et sa capacité de généralisation, la validation est parfois utilisée lors de l'apprentissage (ex: cas du early stopping). Une fois le réseau calculé, il faut toujours procéder à des tests afin de vérifier que notre réseau réagit correctement. Il y a plusieurs méthodes pour effectuer une validation : la cross validation, le bootstrapping... mais pour les tests, dans le cas général, une partie de l'échantillon est simplement écarté de l'échantillon d'apprentissage et conservé pour les tests hors échantillon. On peut par exemple utiliser 60% de l'échantillon pour l'apprentissage, 20% pour la validation et 20% pour les tests. Dans les cas de petits échantillons, on ne peut pas toujours utiliser une telle distinction, simplement parce qu'il n'est pas toujours possible d'avoir suffisamment de données dans chacun des groupes ainsi créent. On a alors parfois recours à des procédures comme la cross-validation pour établir la structure optimale du réseau.

II.9. Les avantages et les inconvénients

Pour les avantages des réseaux de neurones, nous pouvons citer [**Hadjila & Bouabdellah, 03**] :

- Apprentissage automatique des poids (les règles d'apprentissage) ;
- Capacité de généralisation (la phase des tests) ;
- Possibilité de faire le parallélisme (les éléments de chaque couche peuvent fonctionner au parallèle) ;
- Résistance aux pannes (si un neurone ne fonctionne plus, le réseau ne perturbe pas) ;
- Modélisation des fonctions inconnues ;
- Bonne approximation.

Par contre, les réseaux de neurones présentent quelques inconvénients [**Hadjila & Bouabdellah, 03**] :

- Représentation complexe (généralement l'architecture de réseau de neurones est complexe, et même si on la schématise, elle reste peu lisible) ;
- Paramètre difficiles à interpréter (boite noire) ;
- Inexistence des règles définissant l'architecture du réseau en fonction du problème à résoudre ;
- Les algorithmes d'apprentissage consomment un temps de calcul considérable.

II.10. Conclusion

L'intérêt porté aujourd'hui aux réseaux de neurones tient sa justification dans les propriétés fascinantes qu'ils possèdent (et qui sont cités dans le paragraphe sur leurs avantages), ces outils ont permis aussi le dépassement des limites de l'informatique traditionnelle, en offrant des services variés tels que la classification, la modélisation, l'approximation…, tout ces facteurs ont permis à l'informatique de pénétrer plusieurs domaines: le contrôle et la commande automatique, le traitement d'image et du signal numérique en général, la reconnaissance des formes…. Vu ces caractéristiques, nous avons opté pour les réseaux de neurones pour l'amélioration de la qualité des images.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Chapitre III Amélioration de l'image par les réseaux de neurones

III.1. Introduction

La restauration d'images fût ainsi un des premiers problèmes à susciter une grande attention. Celle-ci cherche à corriger les distorsions qui se traduisent lors de l'apparition d'un certain nombre de dégradations, elles se traduisent par une atténuation du contraste, le flou, et au bruit, intervenant lors de la formation des données.

Après avoir vu l'amélioration de la qualité d'image par les techniques de traitement de signal du l'image numérisé, nous allons essayer une autre méthode basée sur les réseaux de neurones pour résoudre le problème.

Le problème étant comment obtenir l'image idéale ou du moins s'en approcher le plus ?

III.2. Intérêt des réseaux de neurones dans le traitement d'image

Les réseaux de neurones ont la spécificité d'apprendre par eux-mêmes à extraire des renseignements cachés dans une masse d'informations, et de fournir ainsi des modèles puissants quant aux connaissances relatives à un problème donné. En d'autres termes, ils représentent une classe d'algorithmes puissants et polyvalents qui sont utilisés pour la classification, la prédiction et le regroupement des données.

Ils ont été ainsi introduits avec succès en traitement d'images, leurs applications sont très nombreuses telles que la détection de contours, la détection d'objets, la segmentation, et la stéréovision.

III.3. Le principe utilisé

Le principe d'utilisation de réseau de neurones en traitement d'image consiste à prendre l'image comme un ensemble d'exemples qui serviront de base d'apprentissage à un réseau de neurones artificiels.

Le filtrage d'une image se fait en déplaçant le réseau sur l'image pixel par pixel en faisant l'extraction du pixel courant et de son voisinage. Pour une image à 256 niveaux de gris, on effectue un changement d'échelle sur les données de la fenêtre pour qu'elle puisse être utilisable par le réseau (données comprises entre 0 et 1).

Soit C la fonction de changement d'échelle telle que $\boldsymbol{C(x) = \frac{x}{255}}$ (pour rendre chaque valeur d'un pixel comprise entre 0 et 1), on obtient alors l'exemple noté P qui servira d'entrée au réseau :

$$\boldsymbol{P = C(I(x, y))}$$

I(x, y) est un pixel de l'image I.

La sortie du réseau contient une cellule donnant le résultat correspondant au traitement désiré. On note S comme étant la fonction qui détermine la sortie du réseau en fonction des entrées de réseau et leurs poids.

$$\boldsymbol{S(P) = O}$$

O : la sortie du réseau.

On effectue ensuite l'opération inverse de la fonction C, c'est-à-dire convertir la sortie du réseau en valeur pour l'image résultat, donc en niveaux de gris :

$$\acute{I}(x,y) = C^{-1}(O) \qquad C^{-1}(C(I(x,y))) = I(x,y) \qquad C^{-1}(x) = 255 * x$$

Donc, en notant par F l'opération du filtre neuronal sur une image I, nous auront:

$$\acute{I}(x,y) = F(I(x,y))$$ I' : L'image Filtré.

Où $$F = C \circ S \circ C^{-1}$$

III.4. Architecture proposée du réseau de neurones

III.4.1. Architecture du réseau

Plusieurs types d'architecture de réseaux et différentes méthodes d'apprentissage ont été utilisés dans le domaine de traitement d'image, parmi celles qui ont obtenue des bons résultats on peut citer le perceptron multicouche et l'apprentissage superviser.

Un Filtre neuronal est un réseau de neurones à configuration classique, c'est-à-dire une configuration organisée en trois couches:

- un ensemble de neurones formant la *couche d'entrée*,
- un second ensemble formant la *couche cachée*,
- un troisième ensemble formant la *couche de sortie*

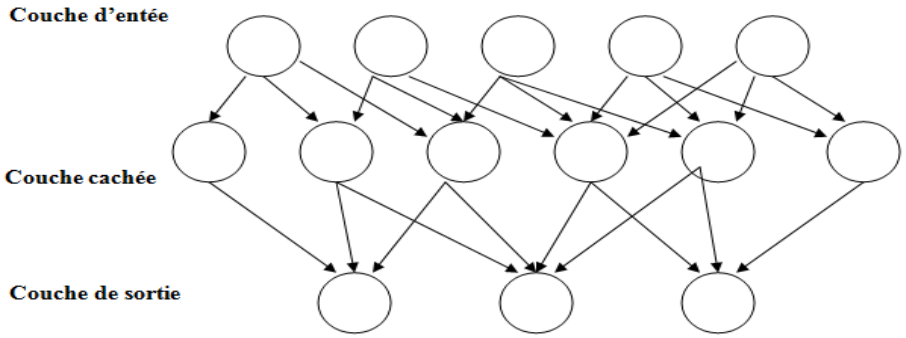


Figure III. 22: *Configuration classique d'un réseau de neurones.*

Le nombre des neurones dans la couche d'entrée varie suivant le nombre des pixels voisinage qui va être prit en compte, c'est-à-dire la taille de fenêtre sur l'image. Cette taille varie suivant le filtre que l'on veut réaliser (les filtres classiques sont en général réalisé par une fenêtre de taille 3×3). Donc pour une fenêtre de taille 3×3, le nombre des neurones dans la couche d'entrée est 9.

Le nombre des neurones pour la couche caché est variable. Le nombre des neurones dans la couche de sortie est 1 car une seule sortie correspondant à la valeur de pixel centrale qui va subir la transformation. L'architecture représentée dans la figure (**Figure III.2**), est celle d'un filtre neuronal pour lequel la fenêtre sur l'image est de taille 3×3.

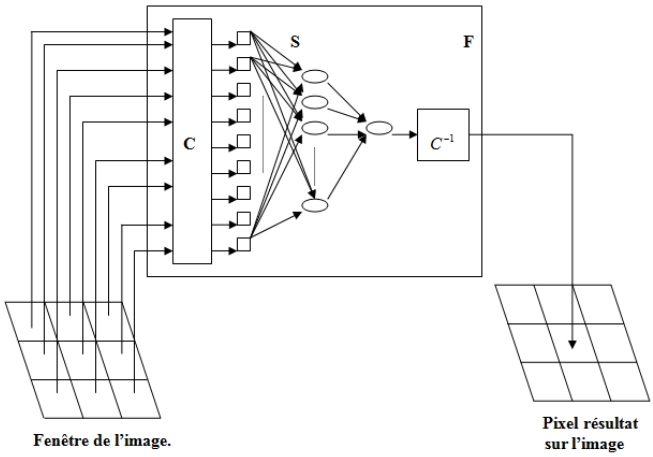


Figure III. 23: Architecture *d'un filtre neuronal.*

III.4.2. Parcours de l'image

Pour que le réseau de neurones puisse apprendre une fonction désirée, il faut lui présenter une série d'exemples appropriés. Dans notre cas, ces exemples seront issus d'images puisque le filtre neuronal doit apprendre à filtrer des images.

Il existe plusieurs méthodes pour extraire les exemples (Pixels) à partir d'une image, parmi celles-ci on cite :

- automatique ;
- aléatoire ;

III.4.2.1. Le parcours automatique

Lors de l'apprentissage et de la phase de filtrage d'une image (reconnaissance), la fenêtre du filtre neuronal est déplacée pixel par pixel sur l'image. Cette architecture permet de filtrer des images de tailles différentes. La figure 4.3 présente le parcours de l'image par le réseau de neurones de manière automatique.

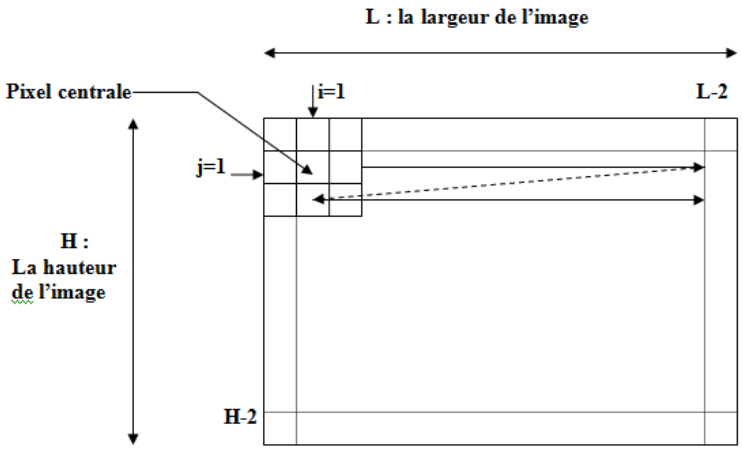


Figure III. 24: *Parcours de l'image.*

Pour un filtre ayant une taille 3*3, le début et l'arrêt du balayage de l'image ce fait comme sur la figure, c'est-à-dire il commence dans (1, 1) et se termine dans (L-2, H-2).

Pour un filtre ayant une taille supérieure, les lignes de départ et d'arrivée ainsi que les colonnes changent. Les bords de l'image ne peuvent pas être modifiés par le filtre neuronal, la fenêtre de balayage devant être identique pour la totalité de l'image.

III.4.2.2. Le parcours aléatoire

Consiste à extraire les exemples des images (Pixels) de manière aléatoire. Cette méthode peut être intéressante car souvent les images comportent des zones homogènes. Ces zones homogènes vont former donc un ensemble d'exemples proches les uns des autres et ceci risque lors de l'apprentissage d'entraîner le réseau inutilement sur le même type d'exemples. Ce choix aléatoire d'exemples de pixels à traiter permet de travailler sur des images de plus grandes tailles et d'obtenir une répartition plus hétérogène des exemples dans l'ensemble d'apprentissage. Dans ce type de parcours, nous avons moins de chance d'avoir des exemples identiques qui se suivent, ce qui augmente la qualité d'apprentissage. Le début et l'arrêt du balayage de l'image reste le même comme pour la méthode précédente de parcours.

III.4.3. Apprentissage

Avant son exploitation, le réseau doit pouvoir déterminer la valeur des poids de chacune de ses connections. Cette valeur est déterminée durant la phase d'apprentissage.

En effet, on appelle apprentissage des réseaux de neurones, la procédure qui consiste à estimer les paramètres des neurones du réseau, afin que ce dernier remplisse au mieux la tâche qui lui est affectée. Dans le cadre de cette définition, on peut distinguer trois classes principales des techniques d'apprentissage : apprentissage supervisé, apprentissage semi supervisé, et apprentissage non supervisé.

Notre filtre neuronal, utilise un apprentissage supervisé qui consiste à présenter un exemple en entrée du réseau, puis de comparer la sortie obtenue avec la sortie désirée.

Ce type d'apprentissage utilise la règle de rétro propagation du gradient. Cette règle permet de déterminer les valeurs des poids dans toutes les couches du réseau en fonction de l'erreur entre la valeur obtenue et la valeur désirée. Cette règle est une des règles les plus utilisées pour le traitement d'images avec des réseaux de neurones multicouches.

Les différents calculs effectués durant l'apprentissage et la reconnaissance sont présentés en détails dans le chapitre précédent Réseaux de neurones.

Lors de l'apprentissage, l'erreur qui est utilisée est l'erreur moyenne quadratique MSE (Mean Square Error) qui est assez utilisée en traitement d'images pour faire une comparaison entre deux images. Cette méthode consiste à effectuer une différence entre chaque pixel de l'image obtenue par le réseau et de l'image désirée. Le calcul se fait avec la formule suivante :

$$\boldsymbol{MSE = \frac{1}{M * N} \sum_{i=1}^{M*N} (x_i - y_i)^2}$$

Où x_i représente un pixel de l'image désirée et yi un pixel de l'image obtenue, M et N étant les dimensions de l'image.

Algorithme général d'apprentissage :

```
   Soit :   E l'erreur minimale que l'on veut obtenir,
            I : Le nombre d'itération,
            N : nombre maximal d'itération.
   I=0 ; erreur=1 ;
   Initialiser les poids ;
   Tant que ((erreur < E) et (I < N)), faire
   Début
      Pour chaque exemple (Pixel) faire
               1. Présenter un exemple en entrée
               2. Propager les signaux d'activation des entrées vers la
                  sortie (Calcul de la sortie)
               3. Comparer la sortie calculée avec la sortie désirée
                  (Calcul de l'erreur)
               4. Rétropropager l'erreur des cellules de sortie vers les
                  cellules d'entrée
               5. Modifier les poids des connections de chaque couche.
      Fin Pour ;
      Incrémenter I ;
Fin.
```

Le calcule du nombre d'exemple prit par le réseau pour chaque image ce fait la manière suivant :

Pour une image de taille 138×104, on a : N= (138 – d1)*(104 – d2)

Avec d1= n1 -1 et d2= n2- 1 avec n1 et n2 largeur et hauteur du filtre.

Si on a un filtre 3*3, N= 13872 exemples.

Ce nombre d'exemples doit être très grand par rapport au nombre de connexions du réseau. Plus ce nombre est grand, plus la généralisation est bonne.

III.4.4. Tests du réseau de neurones

Une fois l'apprentissage effectué, on passe à une étape de généralisation qui consiste à comparer l'image obtenue par le filtre neuronal avec l'image théorique qui à servit à l'apprentissage, puis d'effectuer le même travail avec une image quelconque.

La méthode de tests pour les filtres classique se déroule comme suit : Le réseau devra donc apprendre différents filtres spatiaux. L'avantage principal de ces filtres neuronaux est qu'il pourra filtrer des images pour lesquelles on ne connaît que l'image perturbée et l'image sans perturbations. On pourrait par exemple créer un filtre neuronal devant supprimer le bruit sur une image. Pour cela on va prendre une image nette sur laquelle on va ajouter du bruit de façon aléatoire. Les données d'apprentissage seront donc l'image bruitée en entrée et l'image d'origine en sortie.

III.5. Conclusion

Notre filtre neuronal est très simple, il est basé sur un réseau comportant seulement trois couches (une couche d'entrée, une couche cachée et une couche de sortie), et l'algorithme de rétro propagation de l'erreur. Le filtrage d'une image se fait en déplaçant le réseau sur l'image pixel par pixel en faisant l'extraction du pixel courant et de son voisinage. La sortie du réseau contient une cellule qui donne le résultat correspondant au traitement désiré.

Les avantages des réseaux de neurones étant leur capacité d'apprentissage et de généralisation ainsi que leur architecture les rendant très intéressants pour leur implantation sur machine neuronale et le traitement d'image temps réel.

L'avantage principal de nos filtres neuronaux est qu'il pourra filtrer des images pour lesquelles on ne connaît que l'image perturbée et l'image sans perturbations. On pourrait par exemple créer un filtre neuronal devant supprimer le bruit sur une image. Pour cela on va prendre une image nette sur laquelle on va ajouter du bruit de façon aléatoire. Les données d'apprentissage seront donc l'image bruitée en entrée et l'image d'origine en sortie. Un tel filtre peut être utilisé par exemple pour déparasiter les images qui ont été transmises par ondes hertziennes.

Chapitre IV Implémentation & simulation

IV.1. Introduction

Les méthodes à base de réseaux de neurones constituent une alternative intéressante aux méthodes algorithmiques pour le traitement de données. Au lieu d'expliciter le traitement réalisé par une fonction f au moyen d'un algorithme ou d'un enchaînement d'algorithmes, on construit un programme approximant la fonction f par paramétrage progressif d'un réseau de neurones au moyen de techniques d'apprentissage. Une façon de construire ce programme est de fournir au système d'apprentissage un réseau de neurones initial et un jeu de données à l'aide duquel on fait évoluer le réseau. Le but de ce travail est de développer un tel système en l'appliquant à des problèmes de traitement d'image en général et l'opération de filtrage en particulier.

Ce chapitre présente d'une part l'implémentation de l'application, les différents traitements réalisés sur les images, traitements de base (filtres, segmentation…), et par le réseau de neurones. Et d'une autre part, les tests effectué par les filtres de base et par le réseau de neurones, afin de pouvoir les évalués matière d'amélioration de la qualité de l'image.

IV.2. Environnement de développement

Notre logiciel est écrit en Borland C++ Builder version 6 sous windows. Le choix de la programmation sous Windows a été pris à cause de l'interface graphique qu'offre l'environnement Windows ; le choix du langage Borland C++ Builder 6 était fondamental puisqu'il renferme des outils très puissants pour traiter des données de grande taille telle que les images.

C++ est un outil RAD, c'est à dire tourné vers le développement rapide d'applications (Rapid Application Development) sous Windows. En un mot, C++ Builder permet de réaliser de façon très simple l'interface des applications et de relier aisément le code utilisateur aux événements Windows, quelle que soit leur origine (souris, clavier, événement système, etc.)[2]

C++ Builder repose sur un ensemble très complet de composants visuels prêts à l'emploi. La quasi totalité des contrôles de Windows (boutons, boîtes de saisies, listes déroulantes, menus et autres barres d'outils) y sont représentés, regroupés par famille. Leurs caractéristiques sont éditables directement dans une fenêtre spéciale intitulée éditeur d'objets. L'autre volet de cette même fenêtre permet d'associer du code au contrôle sélectionné.

Il s'agit de programmation « événementielle », des séquences de programme sont exécutées, suite à des actions de l'utilisateur (clique, touche enfoncée, etc.…), détectées par Windows[6].

Notre application est une application de traitement d'image, et le C++ Builder donne des résultats satisfaisants concernant ce thème.

C++ Builder possède un langage de programmation à la fois puissant et simple d'utilisation, il permet d'exprimer les problèmes et les solutions d'une façon aisée.

Il s'impose dans le monde universitaire et industriel comme un outil puissant de réalisation des applications interactives.

[6] www.devellopez.com\c++builder

C++ Builder intègre des outils très puissants pour générer des applications de base de données. Dans notre application nous avons opté pour le SGBD PARADOX lié à Borland C++ Builder 6

PARADOX est un outil qui permet de gérer des bases de données, et offre des services de manipulation et de navigation tels que : l'ajout, la suppression et la mise à jour des informations. Néanmoins, nous n'avons utilisé PARADOX que pour l'implémentation et la manipulation des images enregistrées dans une table.

IV.3. Description de l'application

Notre application est constituée de deux parties distinctes, la première renferme une petite bibliothèque de traitement d'image à savoir : les opérateurs de filtrage (filtre passe-bas, filtre médian…), de segmentation (seuillage, détecteurs de contours).

La deuxième partie est réservée à notre modèle de réseau de neurones qui est décomposée en deux phases :

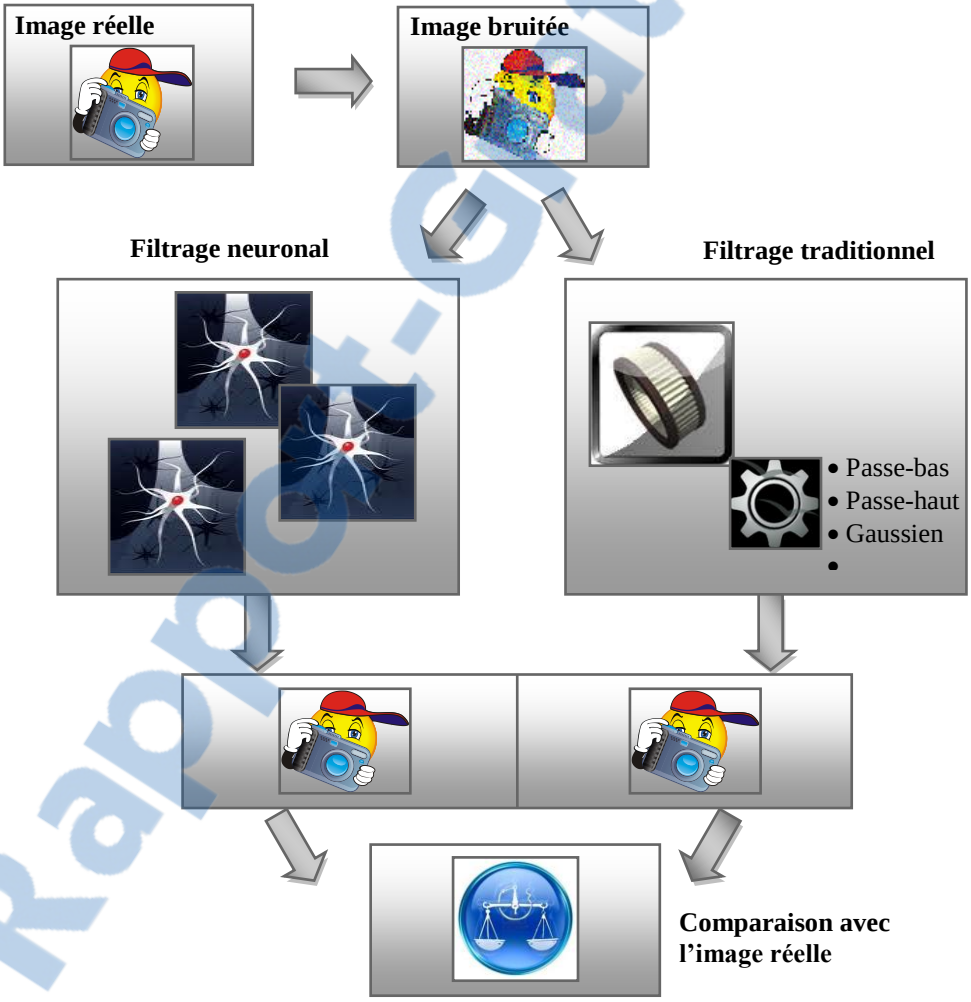


Figure IV. 25: *Principe de fonctionnement.*

- Phase d'apprentissage: elle nécessite la constitution d'une base d'exemples, son but est d'entraîner le réseau de neurones, afin qu'il ajuste ses paramètres (poids).

Rapport-Gratuit.com

- Phase de reconnaissance : elle consiste seulement à présenter les valeurs d'entrée au réseau de neurones qui permettent de donner la valeur de sortie en fonction des poids ajustés en cours de la phase d'apprentissage.

IV.3.1. Partie traitement d'image

IV.3.1.1. Rubrique image

Elle comporte les opérations suivantes:

- Ouvrir: lorsque cet article est activé, la boite de dialogue "Open Picture" apparaît pour sélectionner un fichier image (BMP). Une fois la sélection est validée, l'image est visualisée dans la fenêtre principale en deux types (image couleur RGB et image d'intensité).
- Enregistrer image origine: l'activation de cet article permet de sauvegarder l'image origine sous le format ".BMP".
- Enregistrer image résultat : permet d'enregistrer l'image obtenue par les filtres classiques ou par notre réseau de neurones.
- Effacer: permet d'effacer l'image en cours avec son traitement.
- Ajouter du bruit: permet d'ajouter du bruit à une image, ce bruit peut être de différents types. **Un bruit blanc** qui se caractérise par un certain nombre de pixels répartis aléatoirement sur une image et ayant un niveau de gris maximal (blanc). Le bruit peut aussi apparaître sur l'image sous la forme de pixels ayant des valeurs de niveaux de gris **aléatoires** comprises entre le noir et le blanc et étant répartis de façon aléatoire sur la totalité de l'image.

Image origine

Image bruitée (4% aléatoire)

IV.3.1.2. Rubrique filtres linéaires

Un filtre est linéaire s'il peut s'écrire de la manière suivante :

Soit F le filtre utilisé, I1 et I2 représentant deux images alors :

$$F * (I1+I2) = F*I1 + F*I2.$$

L'opération * représente le produit de convolution.

Cette rubrique contient les filtres linéaires les plus connus comme filtre passe bas, Gaussien et filtre passe haut.

- **Passe- bas**: C'est un filtre qui a des coefficients positifs et leurs somme est égale à un. Il permet d'atténuer le bruit et provoque un effet du flou.

Image bruitée

Filtrage passe-bas

- **<u>Gaussien</u>**: le filtre gaussien donne plus d'importance aux pixels proches du pixel central, et diminue cette importance au fur et à mesure que l'on s'éloigne de celui-ci, il a les mêmes inconvénients que le filtre passe-bas c'est-à-dire il dégrade les contours.

Image bruitée

Filtrage Gaussien

- **<u>Passe-haut</u>**: Il est utilisé pour amplifier les détails de hautes fréquences. Il peut permettre par exemple de restaurer des images qui ont été deffocalisées et d'accentuer les contours en faisant ressortir les pixels compris entre des zones homogènes.

Image bruitée

Filtrage passe-haut

IV.3.1.3. Rubrique filtres non linéaires

Ils sont conçus pour régler les problèmes des filtres linéaires, surtout pour ce qui concerne la mauvaise conservation des contours.

- **<u>Article Médian</u>**: Ce filtre est très utilisé pour éliminer le bruit sur une image qui peut être de différentes origines (poussières, petits nuages, baisse momentanée de l'intensité électrique sur les capteurs, ...) et qui se traduit par des tâches de faible dimension dont la distribution sur l'image est aléatoire.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

Image bruitée

Filtrage Médian

- **Article Minimum**: On applique le même traitement que celui du filtre médian mais la valeur du pixel du centre va être changée par le minimum.

Image bruitée

Filtrage minimum

- **Article Maximum**: On applique le même traitement que celui de filtre minimum, cette fois le pixel du centre va être changé par la valeur maximale.

Image bruitée

Filtrage maximum

IV.3.1.4. Rubrique segmentation

Dans cette rubrique nous trouvons un ensemble de techniques utilisées pour la détection de contours, tel que les détecteurs du premier ordre et de deuxième ordre, on trouve:

- **Article Gradient :** Il trace les contours d'une image.

Image Origine **Image du contour**

- **<u>Article Extraction du contour</u>** : il permet de rendre les contours d'épaisseur d'un pixel.

Image binaire

Image du contour

- **<u>Article Robert</u>** : Cet opérateur est très sensible aux micros contours, mais aussi aux bruits ; les contours sont finis mais incomplets, il est donc appliqué aux régions à structure très fine. La qualité des contours est assez bonne.

Image origine

Image du contour

IV.3.1.5. Rubrique binarisation

Les traitements les plus connus sur les images binaires sont ceux de la morphologie mathématique, mais leurs applications sur les images en niveaux de gris est également possible.

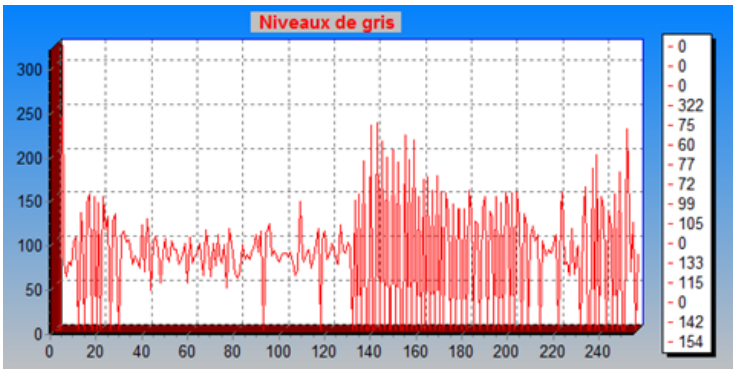
Pour avoir des images binaires à partir d'une image grise, on applique les techniques de seuillage (les valeurs supérieures au seuil seront mises à un (1), et les valeurs inférieures au seuil, à zéro (0)). On trouve sous cette rubrique:

- **<u>Article Histogramme</u>** : Cet article contient un ensemble de techniques de traitement d'histogrammes d'images.
    - *<u>Niveaux de gris</u>* : il permet de donner l'histogramme de l'image courante, qui représente la distribution des niveaux de gris sur les pixels de l'image, les pixels de niveau i sont regroupés et leur somme correspond au nombre d'apparitions du niveau dans l'image.

Image origine

- *Egalisation d'histogramme* : Cette transformation consiste à rendre le plus plat possible, l'histogramme des niveaux de gris de l'image. On désire que chaque niveau de gris soit équitablement représenté dans l'image.

Image origine



- **Article Seuillage** : transforme l'image (en niveaux de gris) en une autre binaire (Noire&Blanc).
  - *Automatique* : permet d'effectuer l'opération de binarisation automatique, le seuil n'est pas choisi par l'utilisateur, il est automatiquement choisi au milieu de l'intervalle [0,255], dans notre cas il est égale à 127.

Image origine

Image binaire

IV.3.2. Traitement par réseaux de neurones

IV.3.2.1. Architecture du réseau de neurones

Le PMC est constitué de 3 couches :

- Le nombre des neurones dans la couche d'entrée varie suivant le nombre des pixels voisinage qui va être pris en compte.

- Le nombre des neurones pour la couche cachée est variable et va être définit durant l'apprentissage du réseau suivant les résultats obtenus.
- Le nombre des neurones dans la couche de sortie est 1 car il s'agit d'une seule sortie représentant la valeur de pixel centrale.

Différents modèles d'apprentissage ont été proposés. On utilisera dans ce travail la méthode de rétropropagation du gradient qui permet de modifier les poids en fonction de l'erreur entre le résultat produit par le réseau et le résultat attendu.

Les poids du réseau sont initialisés de façon aléatoire entre -1 et 1.

L'erreur minimale pour laquelle le PMC termine son apprentissage est fixée à 0.001(on peut dire qu'on a obtenue une image presque identique à l'image désirée), et le nombre maximal d'itérations prévues à l'entraînement est égal à 500 (pour éviter la divergence d'erreur durant l'apprentissage).

IV.3.2.2. Apprentissage du réseau de neurones

Pour notre réseau, nous avons choisi un apprentissage supervisé qui consiste à présenter un exemple en entrée du réseau, puis de comparer la sortie obtenue avec une sortie désirée. Il faut donc, avant de faire l'apprentissage, de donner au réseau l'image origine et l'image désirée, ou filtrée obtenue par un filtre ordinaire classique (filtre passe-bas, sobel, médian….)

Une fois l'apprentissage effectué, on passe à une étape de généralisation qui consiste à comparer l'image obtenue par le filtre neuronal avec l'image théorique qui à servit à l'apprentissage, puis d'effectuer le même travail avec une image quelconque.

IV.3.2.3. La rubrique réseau de neurones

Cette rubrique constitue le cœur de notre travail, elle contient les différentes opérations sur le réseau de neurones (Apprentissage, reconnaissance, création du réseau…).

- *Créer réseau* : il permet de générer un réseau de neurone avec une configuration choisie par l'utilisateur. Lorsque cet article est activé la fenêtre « Création du réseau », apparaît pour donner le nom du réseau et le nombre des couches qui doit être compris dans l'intervalle [3,7].
  Une fois le choix est validé, une autre fenêtre apparaît pour donner le nombre des neurones pour les couches cachées.
  Le nombre des neurones dans la couche de sortie est 1, mais ce nombre dans la couche d'entrée varie suivant le nombre des pixels voisinage qui va être pris en compte (c'est-à-dire suivant la taille du masque).
  Les poids du réseau sont initialisés de façon aléatoire entre -1 et 1.
- *Charger réseau* : Il permet de charger un réseau déjà existant à partir d'une unité de stockage (disque dur notamment). L'utilisateur choisi le nom du réseau et le programme le charge pour pouvoir l'exploiter ensuite dans des opérations disponibles.
- *Enregistrer réseau* : Permet de sauvegarder le réseau courant.
- *Apprentissage* : Nous sommes en présence deux types d'apprentissage :
  - **Automatique** : Consiste à extraire les exemples à partir d'une image de manière automatique (voir chapitre filtrage d'image par réseau de neurones).
  - **Aléatoire** : Consiste à extraire les exemples de manière aléatoire.

Rapport-gratuit.com
LE NUMERO 1 MONDIAL DU MÉMOIRES

- *Reconnaissance*: Il permet de faire le traitement du réseau courant sur l'image qui existe dans la fenêtre principale7.
- *Résumé d'apprentissage* : donner l'architecture du réseau courant et la variation d'erreur durant l'apprentissage.
- *Article Charger image* : lorsque cet article est activé la fenêtre « Gestion des images » apparaît pour choisir une image à partir de la base de données, une fois le choix est validé, l'image est visualisée dans la fenêtre principale.

IV.3.3. Phase des tests & résultats

Apres la présentation de l'architecture de notre système ainsi que l'algorithme d'apprentissage que nous avons choisi, nous présentons dans ce qui suit la procédure de tests que nous avons élaborée.

En effet, on se propose de montrer les principaux résultats de restauration en appliquant la méthode proposée sur des images dégradées obtenues sur l'image originale en appliquant le filtre de base et le réseau de neurone. Pour une meilleure lisibilité, nous avons déroulé la procédure de test seulement sur un filtre passe-bas, et l'apprentissage du réseau de neurone a était effectué sur une image bruitée comme entrée du réseau et l'image réelle comme sortie désirée de ce dernier.

Déroulement des tests :

1. Chargement de l'image réelle (*Img_R*) et application d'un bruit sur cette dernière (*Img_B*) ;
2. Application du filtre passe-bas sur *Img_B*, le résultat est une nouvelle image filtrée *Img_F*;
3. Apprentissage du réseau de neurones, comme entrée nous avons *Img_B (*l'image bruité) et comme sortie l'image *Img_R*;
4. Test de l'image *Img_B* par le réseau de neurones ce qui nous donne l'image *Img_RN*
5. Calcule de l'erreur quadratique moyenne normalisée entre *Img_R* (l'image réelle) et l'image *Img_F* , ensuite avec l'image résultat du réseau de neurone (*Img_RN*)

   L'erreur quadratique moyenne normalisé : $\boldsymbol{NMSE} = \frac{\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}[f(i,j)-\grave{f}(i,j)]^2}{\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}[f(i,j)]}$
6. Répétition des étapes 1, 2, 4, 5 sur une nouvelle image *Img_N* (c'est-à-dire on compare le résultat du filtre neuronale de la nouvelle image bruitée avec l'image réelle sachant que l'apprentissage de réseau à était déjà effectué sur la première *Img_R*)

Premier test : Apprentissage du réseau de neurones effectué sur une image bruité de 5%

Image	Bruit de couleur aléatoire	Filtre passe bas	Restauration par réseau de neurone
		NMSE	NMSE
Image 1	**3 %**	**0.0288**	**0.0400**
	5 %	**0.0290**	**0.0409**
	10 %	**0.0313**	**0.0425**
Image 2	**3 %**	**0.006**	**0.0226**
	5 %	**0.008**	**0.0244**
	10 %	**0.009**	**0.0260**

[7] *La reconnaissance se fait après l'apprentissage*.

Deuxième test : Apprentissage du réseau de neurones effectué sur une image bruité de 10%

Image	Bruit de couleur aléatoire	Filtre passe bas	Restauration par réseau de neurone
		NMSE	NMSE
Image 1	3 %	0.0287	0.0393
	5 %	0.0297	0.0400
	10 %	0.0318	0.0418
Image 2	3 %	0.008	0.0602
	5 %	0.008	0.0584
	10 %	0.012	0.0560
Image 2	3 %	0.025	0.0508
	5 %	0.025	0.0511
	10 %	0.027	0.0517

Troisième test : Apprentissage du réseau de neurones effectué sur une image avec bruit aléatoire

Image origine

Image bruité

Image améliorée par réseau de neurones
NMSE = 0.063

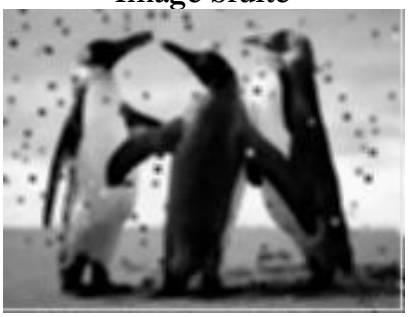


Image améliorée par filtre passe-bas
NMSE = 0.1013

IV.3.4. Conclusion

Nous avons présenté dans ce chapitre les principales fonctions de l'application pour l'amélioration de la qualité d'images, nous avons parlé des filtres de base et le traitement par le réseau de neurones multi couches, l'idée de base est de fournir des outils pour l'amélioration de la qualité de l'image par le réseau de neurones, ou par les filtres de base.

Conclusion Générale et Perspectives

Depuis l'apparition des réseaux de neurones en informatique vers les années 40, ils ont déjà été utilisés dans un grand nombre d'applications très diverses et ont permet d'obtenir de meilleurs résultats à certains types de problèmes que ceux obtenus par les autres méthodes

Les exemples utilisant les réseaux de neurones les plus connus sont NetTalk, pour la diction de textes écrits en anglais. Ils ont été aussi utilisés en robotique pour par exemple apprendre à marcher à un robot à six pattes, ou dans des jeux de vie artificielle, comme le jeu bien connu du grand public, Créatures, utilisant aussi les algorithmes génétiques.

Dans le traitement d'image, les réseaux de neurones sont aussi de plus en plus utilisés. Lors de la réalisation des différentes applications, telle que la compression, la détection de contours, l'extraction de régions, la reconnaissance de textures, la reconnaissance de caractères et la vision stéréoscopique, les réseaux de neurones ont permet d'obtenir de meilleurs résultats. Dans notre projet nous avons exploité cette puissance et cette souplesse des réseaux de neurones pour concevoir et implémenter un réseau de neurones multi couche pour l'amélioration de la qualité d'image bruitée. Dans l'ensemble, les résultats obtenus par le réseau de neurones dans le domaine de traitement d'images sont intéressants.

Ce travail ouvre la voie vers diverses perspectives de recherche, l'utilisation d'autres types de réseaux de neurones tel que Hopfield, pour la restauration d'images, permettant peut être d'avoir de meilleurs résultats, dans la restauration d'images.

Références Bibliographiques

[**Besançon, 98**]
Besançon B, « *vision par ordinateur en deux et trois dimension* », Edition Eyrolles, 1998.

[**Burel, 01**]
Burel G., « *Introduction au traitement d'images* », Edition Hermès, 2001.

[**Cocquerez & Philipp, 98**]
Cocquerez J.-P., Philipp S., « *Analyse d'images : filtrage et segmentation* », Edition Masson, 1998.

[**Devaux, 97**]
Devaux F., « *Filtrage d'images par réseaux de neurones* », 1997.

[**El Hannani, Koudacke, 2002**]
El Hannani F., Koudacke L, « *Élaboration d'un système de détection et de reconnaissance des visages & identification des émotions faciales* », Université de Sidi Bel Abess, 2002.

[**Gonzales & Wintz, 77**]
Gonzales R., Wintz P. « *Digital Image Processing* », Addison Wessley, 1977.

[**Hadjila & Bouabdellah, 03**]
Hadjila F. & Bouabdellah R., « *Reconnaissance des visages par les réseaux de neurones* », Mémoire de PFE, Université de Tlemcen, 2003.

[**Jadouin, 94**]
Jadouin J. F., « *Les réseaux neuromimétiques* », Edition Hermes, 1994.

[**Naciri & Chaoui, 03**]
Naciri H., Chaoui N., « *Conception et Réalisation d'un système automatique d'identification des empreintes digitales* », Mémoire de PFE, Université de Tlemcen, 2003.

[**Tabari & Tagma, 96**]
Tabari K., Tagma S., « *Compression d'images animées à très faible débit par la géométrie des fractales* », Projet de fin d'études, USTBH, 1996.

Rapport-Gratuit.com

Résumé

La réduction du bruit et du flou est une tâche très importante en traitement d'images. Pour résoudre ce problème de restauration d'images bruitées, notre choix s'est porté sur le réseau de neurones perceptron multicouches. L'idée de base est de montrer les performances des réseaux de neurones et leurs capacités de généralisation dans le traitement d'images.

L'évaluation des résultats obtenus, à l'aide du calcule de l'erreur (NMSE L'erreur quadratique moyenne normalisé), suscite un intérêt particulier pour la reconnaissance d'images par les réseaux de neurones.

Mots Clés : Réseaux de neurones, filtres de base, traitement d'image.

Abstract

Noise reduction and blur is a very important task in image processing. To resolve this problem of restoring noisy images, we chose the multi-layer perceptron neural network. The basic idea is to show the performance of neural networks and their ability to spread in the image processing.

The evaluation of the results obtained was madding by using (NMSE) method, raises a particular interest in image recognition by neural networks.

Keywords: Neural networks, basic filters, image processing.

ملخص

التقليل من التشويش و عدم وضوح الصور هي عميلة هامة في ميدان المعالجة الآلية للصور،من أجل تحسين الصور ومحاولة تقريبها إلى حالتها الطبيعية ، قمنا باستعمال شبكة العصبية المتعددة الطبقات ، وهدا لإمكانيات هده الأخيرة في هذا المجال.

ولتقدير الخطأ الوارد في هذه العملية قمنا بالاستعانة بمعادلة حساب الخطأ (NMSE)

الكلمات الرئيسية: الشبكات العصبية، مرشحات الأساسية، المعالجة الآلية للصور.