

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE D'ORAN



MOHAMED BOUDIAF-



FACULTE DE MATHÉMATIQUE et INFORMATIQUE

DÉPARTEMENT D'INFORMATIQUE

Mémoire présenté par

Abdelaziz BELAID

Pour l'obtention du diplôme de **MAGISTER** en **INFORMATIQUE**

Reconnaissance de l'écriture en ligne par les réseaux de
neurones dynamiques

Option

Reconnaissance des Formes - Intelligence Artificielle

(RF-IA)

soutenue publiquement le : 21/01/2015 devant le jury:

Président	Mr Mohamed BENYETTOU	Professeur	USTO-MB
Rapporteur	Mr Abdelkader BENYETTOU	Professeur	USTO-MB
Co-encadreur	Mr Redouane TELMSANI	MCB	INTTIC Oran
Examinatrices	Mlle Nacera BENAMRANE	Professeur	USTO-MB
Examinateur	Mr Abderahime BELMADANI	MCA	USTO-MB

Années universitaire 2014/2015

Résumé

Le cadre de ce travail se situe dans le domaine de la reconnaissance de caractères arabes isolés en ligne, et la motivation principale dans celui-ci est d'étudier le manuscrit arabe via à vis la technologie en ligne. Notre système s'appuie sur un réseau de neurones à convolution pour la reconnaissance de caractères arabes isolés, pierre angulaire de nos travaux futurs sur le reconnaisseur mot. Nous détaillerons le système implémenté basé sur les réseaux de neurones à décalage temporel (TDNN), cette approche propose une architecture composée de deux parties, la première partie est une phase d'extraction des caractéristiques propre au TDNN, la deuxième partie est un perceptron multicouches (PMC) classique élaboré pour la classification. Pour les besoins de cette application, nous ferons appel à notre base de données NOUN. Le paramétrage de ces deux parties nous permettra d'analyser l'impact de la topologie TDNN sur les résultats de taux de reconnaissance des caractères.

Les scores pour TDNN avec les algorithmes d'apprentissages RP, LM, GD et SCG sont respectivement 68,49 % , 79,97% , 30,53% et 62,52% à base de reconnaissance.

Mots-clés : Reconnaissance des formes, Reconnaissance de Caractères Isolés en ligne, Manuscrit Arabe, Réseaux de Neurones à Décalage Temporel.

Abstract:

The frame work is located in the field of the character isolated Arabic recognition on line. The principal motivation for this work is to study the Arabian manuscript and the on line technology. Our system is based on a neural network for isolated Arabic character recognition, which is the first step toward our future works on the word recognition. This system is based on the time delay neural network (TDNN). Our approach proposes an architecture composed of two parts, the first part is an extraction phase of the features in the TDNN, the second is a perceptron multilayer (PMC) elaborated for the classification. For the requirement of this application, we designed a corpus named NOUN data base. The parameterization of these two parts will allow to analyze the impact of the TDNN topology on characoriter recognition rate.

The scores for TDNN with the algorithme apprenticeship are respectively 68,49 % , 79,97% , 30,53% and 62,50% a significant recognition

Keywords: Pattern recognition, On line isolated character recognition, Arabic Manuscript, Time Delay NeuralNetwork.

ملخص

يقع هذا العمل في مجال التعرف الآلي المباشر على الحروف العربية، و الدافع الأساسي هو دراسة الكتابة العربية تجاه التكنولوجيا المباشرة.

نظامنا يعتمد على شبكة النورونات العصبية للتعرف الآلي المباشر على الحروف العربية المعزولة ويعتبر حجر الاساس في عملنا المستقبلي اي التعرف على الكلمة. و سوف نتطرق بالتفصيل الي نظام تنقيذها على أساس الشبكات العصبية (TDNN)، وهذا النهج يوفر بنية مكونة من جزأين ، الجزء الأول هو مرحلة الاستخراج التي تميز TDNN ، والجزء الثاني هو برسبترون متعدد الطبقات (PMC) ، وضع لمعيار التصنيف ، وسوف نستخدم قاعدة بيانات المعلومات الخاصة بالمخبر "نون".
ترقيم هذين الجزئين سوف يسمح لنا بتحليل أثر طوبولوجيا TDNN على نتائج معدل التعرف على الحروف

نتائج البحث باستعمال TDNN الذي استعملت فيه خوارزميات التدريب RP LM GD SCG أعطت النسب على التوالي:

. 62,52% 30,53% 79,97% 68,49 %

الكلمات المفاتيح

التعرف الآلي على الأشكال، التعرف الآلي المباشر على الحروف ، الخط العربي، شبكة النورونات

TABLE DES MATIERES

Introduction Générale	10
I.CHAPITRE I: INTRODUCTION AU SIGNAL D'ECRITURE	12
<i>I.1. Introduction</i>	12
<i>I.2 Reconnaissance en-ligne\ hors-ligne</i>	15
I.2.1 L'écriture en ligne	15
I.2.2 L'écriture hors-ligne	17
I.2.3 Reconnaissance	19
I.2.4 Vérification des signatures	20
I.2.5 Quelques exemples d'applications	22
I.2.6 Les différents types de documents	23
<i>I.3 Principes généraux sur l'écriture en-ligne</i>	24
I.3.1 Formalisation de la reconnaissance de formes	25
I.3.1.1 Les entrées du classifieur	25
I.3.1.2 Modélisation intrinsèque et modélisation discriminante	27
I.3.1.3 Les sorties du classifieur	29
<i>I.4 Conclusion</i>	30
II.CHAPITRE II: RECONNAISSANCE DE L'ECRITURE ARABE MANUSCRITE	31
<i>II.1 Problématique</i>	31
II.1.1 Présentation de la langue arabe	31
II.1.2 Caractéristiques morphologiques de l'écriture arabe	33
II.1.2.1 Signes diacritiques	37
II.1.2.2 Une ou plusieurs composantes connexes par mot	38
II.1.2.3 Ligatures verticales	39
<i>II.2 Etat de l'art : Bases de données & approches de classification</i>	39
II.2.1 Principales bases de données existantes	39
II.2.1.1 Kharma/Ahmed/Ward 1999	40
II.2.1.2 AHDB	40
II.2.1.3 CENPARMI	41
II.2.1.4 IFN/ENIT	41
II.2.1.5 CEDARABIC	42
II.2.1.6 Base de données latine UNIPEN	43
II.2.1.7 Base de données latine IRONOFF	44
II.2.1.8 Base de données arabe ADAB	45
II.2.2 Différentes systèmes existants:	45
II.2.2.1 Surveys existants	45
II.2.2.1.1 ICDAR 1997	45
II.2.2.1.3 PAMI 2006	46
II.2.2.1.4 SACH 2006	47

II.2.2.1.5 ISSPA 2007	48
<i>II.3 Différentes approches de reconnaissance</i>	50
II.3.1 Approche structurelle	50
II.3.2 Méthodes des k plus proches voisins (kppv)	51
II.3.3 Les machines à vecteurs supports (SVM)	51
II.3.4 Méthodes connexionnistes	52
II.3.5 Méthodes stochastiques	53
II.3.5.1 Modèles de Markov cachés (HMM)	54
II.3.5.2 Réseaux Bayésiens statiques et dynamiques	54
II.3.6 Les contributions originales	55
<i>II.4 Conclusion</i>	50
III.Chapitre III: les réseaux de neurones à convolution (TDNN)	56
<i>III.1 Introduction:</i>	56
<i>III.2 Les réseaux de neurones artificiels(RNA)</i>	56
III.2.1 Le neurone formel	56
III.2.1.1 Définition d'un RNA	58
III.2.2 L'architecture de base : le perceptron multi couches (PMC)	58
III.2.2.1 Structure d'un PMC	58
III.2.2.2 Apprentissage	59
III.2.2.2.1 La Rétro Propagation du Gradient	60
III.2.2.2.2 Rétropropagation du gradient "résistante" : Rprop	62
III.2.2.2.3 Méthode de Levenberg-Marquardt	63
III.2.2.2.3 Méthode d'apprentissage par algorithme du gradient conjugué	65
III.2.2.3 Le paramétrage	65
III.2.2.3.1 Initialisation des poids et choix du pas	65
III.2.2.3.2 Architecture d'un réseau de neurones	65
<i>III.3 Les réseaux de neurones à convolution</i>	66
III.3.1 Réseaux de neurones adaptés aux données spatiotemporelles	67
III.3.1.1 Représentation spatiale de l'information temporelle	68
III.3.1.2 Les limites de la représentation spatiale	69
III.3.1.3 Classification d'architectures de réseaux de neurones\temps	69
<i>III.4 TDNN réseau de neurone à délais temporels</i>	71
III.4.1 Introduction	71
III.4.2 Propriétés du TDNN	71
III.4.3 TDNN et Apprentissage	72
III.4.4 L'unité temps-retard	74
III.4.5 Principe des poids partagés	76
III.4.6 Architecture générale du TDNN	77
<i>III.5 Conclusion</i>	80

IV.Chapitre IV:IMPLEMENTATION, RESULTATS ET DISCUSSION	81
<i>IV.1 Introduction</i>	81
<i>IV.2 Présentation d'un reconnaisseur de caractères isolés</i>	81
<i>IV.3 Construction de la base de données NOUN</i>	82
IV.3.1 Dispositifs matériels	82
IV.3.1.1 Ordinateur	82
IV.3.1.2 Tablette PC	82
IV.3.2 Fiche d'écriture Scripteur	85
IV.3.2.1 Alphabet	85
IV.3.2.2 Interface d'acquisition	86
IV.3.3 Description de la base	86
<i>IV.4 Expérimentation et Résultats</i>	87
IV.4.1 Prétraitement des caractères	87
IV.4.1.1 Échantillonnage spatial	87
IV.4.1.2 Extraction des caractéristiques	88
<i>IV.5 Topologie et paramètres</i>	91
IV.5.1 Architecture requise du TDNN	91
IV.5.2 Aperçu sur le modèle	92
<i>IV.6 Expérimentation et Résultats</i>	94
IV.6.1 expérience 1(caractères similaires)	94
IV.6.2 Discussion	98
IV.6.3expérience 2(caractères différents)	98
IV.6.4 Discussion	101
IV.6.5expérience 3: (Apprentissage des 28 caractères)	101
IV.6.6 Discussion	103
<i>IV.7 Phase de généralisation</i>	103
<i>Conclusion générale et perspectives</i>	105
<i>Bibiographie</i>	107

Table des Figures :

<i>Figure I-1: Exemples de systèmes nomades avec leur système de reconnaissance de l'écriture manuscrite.</i>	13
<i>Figure I-2: Principe de la reconnaissance d'écriture manuscrite.</i>	14
<i>Figure I-3: Importance de la prise en compte du contexte des mots.</i>	15
<i>Figure I-4: Systèmes d'acquisition d'écriture en ligne.</i>	16
<i>Figure I-5: Représentations du signal du mot "age".</i>	18
<i>Figure I-6: Schéma général d'un système de reconnaissance d'écriture manuscrite.</i>	20
<i>Figure I-7: Une signature authentique (a) avec des échantillons imités (b, c et d) extraits à partir de la base de test.</i>	22
<i>Figure I-8: Deux types de modélisation : (a) discriminante et (b) intrinsèque</i>	28
<i>Figure II-1: Exemples d'écritures arabes reflétant ses caractéristiques morphologiques.</i>	35
<i>Figure II-2: Sens de l'écriture des lettres arabes</i>	36
<i>Figure II-3: Quelques styles de Calligraphie arabe</i>	36
<i>Figure II-4: Changement de prononciation de la même forme selon la position des points</i>	37
<i>Figure II-5: Voyelles en arabe : (a) A, (b) OU, (c) I, (d) ~, (e) AN, (f) OUN, (g) IN.</i>	38
<i>Figure II-6: Autres signes diacritiques : (a) hamza, (b) chadda, (c) madda</i>	38
<i>Figure II-7: (a) Le mot « drapeau » est composé d'un pseudo-mot, (b) Le mot « canapèche » est composé de 3 pseudo-mots.</i>	39
<i>Figure II-8: Ligatures verticales et inversion de l'ordre du tracé</i>	39
<i>Figure II-9: Exemple de chèque saoudien, et son montant littéral.</i>	41
<i>Figure II-10: Deux images et leurs annotations respectives dans la base IFN/ENIT. B pour Begin (début), M pour Middle (milieu), A pour Alone (isolé) et F pour Final (finale).</i>	42
<i>Figure II-11: Exemple de segmentation en pseudo-mots qui porte à confusion .</i>	42
<i>Figure II-12: Modèle de McClelland et Rumelhart</i>	47
<i>Figure II-13: Séparation souple par SVM: marge et hyperplan séparateur</i>	51
<i>Figure II-14: Topologie d'un réseau de neurones.</i>	53
<i>Figure III-1: le modèle du neurone formel</i>	57
<i>Figure III-2: architecture de base d'un PMC</i>	59
<i>Figure III-3 : Algorithme de iRprop+</i>	63
<i>Figure III-4: L'algorithme de Lavenberg-Marquard</i>	64
<i>Figure III-5: Illustration des connexions dans un PMC et dans un RNC</i>	67
<i>Figure III-6: Représentation spatiale de l'information en entrée d'un réseau de neurones</i>	68
<i>Figure III-7: Classification d'architectures de réseaux de neurones</i>	70
<i>Figure III-8: Ajout d'une dimension temporelle par introduction de la notion de retard</i>	75
<i>Figure III-9: Représentation du TDNN</i>	77
<i>Figure III-10: Architecture du TDNN pour l'écriture</i>	78
<i>Figure IV-1: Schéma général d'un SRC</i>	81
<i>Figure IV-2: Fonctionnalités de la tablette WACOM</i>	83
<i>Figure IV-3: Composants de stylets et réglage de paramètres</i>	83
<i>Figure IV-4: Positionnement de stylet</i>	84
<i>Figure IV-5: Zone de travail sur tablette</i>	84
<i>Figure IV-6: Alphabet saisie par le scripteur</i>	85

<i>Figure IV-7: Interface d'acquisition</i>	86
<i>Figure IV-8: Représentation du processus d'extraction en-ligne</i>	87
<i>Figure IV-9: Algorithme d'échantillonnage spatial</i>	88
<i>Figure IV-10: Caractère "Noun" avant et après échantillonnage spatial</i>	88
<i>Figure IV-11: Illustration des angles de la direction (θ) et de courbure(Φ)</i>	89
<i>Figure IV-12: Algorithme de normalisation et d'extraction des caractéristiques</i>	89
<i>Figure IV-13: L'extraction des caractéristiques de la lettre Noun</i>	90
<i>Figure IV-14: TDNN avec une fenêtre temporelle de taille 4 reliée à chaque neurones de la couche caché</i>	92
<i>Figure IV-15: TDNN à 1 couche cachée</i>	92
<i>Figure IV-16: Aperçu du model</i>	93
<i>Figure IV-17: Découpage des neuf groupes d'apprentissage et illustration de la ressemblance entre caractères</i>	93
<i>Figure IV-18: Architecture idéale du réseau TDNN</i>	95
<i>Figure IV-19: Sortie du réseau, EQM pour les 9 groupes</i>	96
<i>Figure IV-20: Les matrices de confusion et de performance pour chaque groupe</i>	97
<i>Figure IV-21: les taux de généralisation des caractères des neuf groupes</i>	98
<i>Figure IV-22: Découpage des trios groupes d'apprentissage et illustration de la différents entre caractères.</i>	99
<i>Figure IV-23: Les matrices de confusion et de performance pour chaque groupe</i>	100
<i>Figure IV-24 : les taux de généralisation des caractères des trios groupes</i>	100
<i>Figure IV-25 : Découpage des quatre groupes d'apprentissage</i>	101
<i>Figure IV-26: les taux de généralisation des 28 caractères avec quatre algorithmes d'apprentissage</i>	102
<i>Figure IV-27: Test du caractère en ligne "Noun"</i>	105

Liste des tableaux

<i>Tableau II-1: Les lettres arabes</i>	32
<i>Tableau II-2 : Variabilité des styles d'écriture des points</i>	37
<i>Tableau II-3: Taille de sous-corpus de la base UNIPEN</i>	43
<i>Tableau II-4 : Description de la base de données UNIPEN</i>	44
<i>Tableau II-5: Taille de sous-corpus de la base IRONOFF</i>	44
<i>Tableau II-6: sous-corpus de la base de données ADAB 1,2 et 3</i>	45
<i>Tableau IV-1: Les taux de C, EQM et le temps d'exécution pour chaque groupe</i>	97
<i>Tableau IV-2: Les taux de C, EQM et le temps d'exécution pour chaque groupe</i>	99
<i>Tableau IV-3: Résultats des variations de la fonction d'apprentissage</i>	102

INTRODUCTION GENERALE

Même si les ordinateurs envahissent le monde de la communication, l'écrit et la parole demeurent deux modalités de communications privilégiées. A un horizon plus ou moins éloigné, nombreux travaux et prospectives cherchent à faire disparaître l'ordinateur de l'environnement de l'utilisateur. Néanmoins, celui-ci sans qu'il en ressente les contraintes sera connecté aux systèmes d'information par des interfaces naturelles: un geste de la main, un regard, la parole... et bien entendu l'écriture manuscrite. Du vieux concept utopique du bureau sans papier, on arrive sur le paradigme du bureau sans ordinateur perceptible.

Aujourd'hui, c'est encore bien souvent l'humain qui fait les efforts d'adaptation. Les mini-claviers des téléphones portables en sont un exemple. Leur ergonomie est assez limitée dès lors que l'on cherche, par exemple, à composer un petit message. Pourtant, la tendance est nette ; des progrès importants ont été réalisés pour rapprocher le monde du numérique et celui de l'écriture manuscrite. L'apparition des assistants personnels (PDA), des tablettes PC, des smart-phones, ces téléphones dernières générations qui combinent une multitude de fonctionnalités -agendas personnels, bloc-notes, jeux vidéos, appareil photo et caméra... - confirme cette situation.

Des progrès très significatifs ont également été enregistrés dans le confort d'usage des stylos digitaux. Ils sont devenus aujourd'hui très comparables aux stylos usuels : leur poids et leur volume ont été divisés par deux en trois ans. Il est ainsi possible d'écrire avec un stylo quasi-ordinaire, sur du papier, et de traiter en temps-réel cette information éventuellement n'importe où sur la planète.

Introduction Générale

En parallèle à tous ces progrès matériels, il reste à mieux maîtriser l'interprétation de ces traces écrites qui vont proliférer sur de multiples dispositifs. C'est dans le cadre de ces applications de saisie d'écriture en-ligne à capacités limitées que s'inscrivent nos travaux.

CHAPITRE I: INTRODUCTION AU SIGNAL D'ECRITURE

I.1. INTRODUCTION

L'écriture, transcription de la parole ou de la pensée par des signes ou des caractères normalisés, est apparue il y a plus de 5000 ans en Mésopotamie. L'important pour les peuples de l'époque était de pouvoir fixer des lois et règles de vie et d'en transmettre des enseignements à leurs descendants. L'écriture fût, alors, une modalité de communication utilisée par les humains depuis bien longtemps pour se transmettre des informations d'une époque à une autre. L'écriture est devenue, aussi, un complément indispensable à la parole, d'une part parce qu'elle assure le stockage et la mémorisation et d'autre part parce qu'elle est mieux adaptée que la parole pour certaines tâches.

Les progrès réalisés depuis quelques années dans les domaines de l'informatique, de l'électronique et des systèmes de communication, ont engendré de très nombreuses applications reliées à l'écriture qui se sont intégrées de plus en plus rapidement à notre travail quotidien. Aujourd'hui, l'ordinateur fait partie intégrante de ce nouvel environnement, parce qu'il nous assiste dans diverses disciplines et pour de tâches multiples. Ceci a rendu nécessaire le développement de modalités de communication plus performantes entre humain et machine.

Donc, la reconnaissance de l'écriture manuscrite n'est pas un sujet nouveau, il remonte à plus d'une trentaine d'années. La reconnaissance de l'écriture manuscrite s'avère un problème extrêmement complexe qui n'a pas de solution satisfaisante à ce jour. Ainsi rapidement, les chercheurs ont restreint leurs études à des problèmes particuliers en liaison avec des applications bien définies. Parmi celles-ci on cite les applications de la reconnaissance de l'écriture manuscrite : le tri automatique du courrier postal, le traitement automatique de dossiers administratifs, des formulaires d'enquêtes, ou encore l'enregistrement des chèques bancaires.

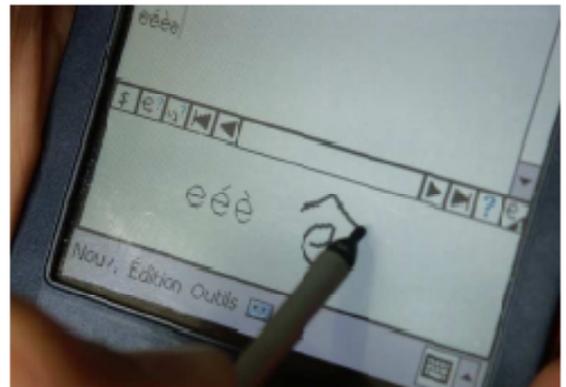
L'émergence de l'informatique nomade a rendu indispensable l'utilisation des interfaces orientées stylo : l'utilisateur écrit et dessine directement des textes, des croquis, des

commandes, etc sur l'écran tactile du système, l'interface classique clavier/souris étant à la fois plus encombrante et moins adaptée à l'utilisation de ces périphériques. Ce type d'interface a de nombreux avantages dont celui d'être accessible dans de nombreuses situations (nécessité de déplacement, bruit, station debout...) dans lesquelles les utilisateurs peuvent se trouver (contrôle de chantier, médecin à l'hôpital...). De plus les interfaces stylo permettent de réduire la taille du périphérique puisque le clavier n'est plus nécessaire. Cette thèse se place dans ce contexte général d'utilisation de périphériques mobiles de petite taille et donc ayant peu de ressources disponibles en terme de puissance de calcul et de mémoire : téléphone intelligent (Smartphone), assistant personnel (PDA)...

La figure I.1 présente deux types de périphériques mobiles avec chacun une interface différente pour la reconnaissance de l'écriture. [1]



Un Smartphone



Un PDA

Figure I-1: Exemples de systèmes nomades avec leur système de reconnaissance de l'écriture manuscrite.

Dans ces exemples, il s'agit de systèmes de reconnaissance de caractères. En effet, chaque caractère est reconnu séparément des autres caractères. De plus, dans ce type de contexte applicatif, la reconnaissance se fait à la volée, c'est-à-dire au fur et à mesure de la saisie des caractères par l'utilisateur.

Afin de pouvoir tirer partie des documents ainsi produits, il est nécessaire de proposer des méthodes de traitement adaptées, parmi lesquelles se trouve la reconnaissance d'écriture manuscrite. La reconnaissance d'écriture manuscrite [2] consiste à transformer un signal numérique en sa représentation symbolique (voir figure I.2).



Figure I-2: Principe de la reconnaissance d'écriture manuscrite.

Ce mémoire se place dans le cadre de la reconnaissance d'écriture manuscrite et plus particulièrement dans celui de la reconnaissance de caractères isolés, saisis en-ligne sur des périphériques tels que des Tablets PC.

La reconnaissance d'écriture manuscrite est une tâche complexe, à-cause de la variabilité des styles d'écriture. La difficulté de cette tâche est d'autant plus importante au niveau de la reconnaissance de textes puisqu'il s'agit d'une des étapes ultimes de la reconnaissance. En effet, la reconnaissance de textes s'appuie sur de nombreux processus pour faire face à un enchaînement de problématiques allant de la segmentation de la phrase en mots à la reconnaissance de mots, en passant par la reconnaissance de lettres. . . Dans tous ces processus, le contexte joue un rôle fondamental pour réussir à interpréter correctement les différentes hypothèses de reconnaissance de formes.

Nous pouvons distinguer la reconnaissance de l'écriture manuscrite, qui s'appuie uniquement sur la reconnaissance de formes, de son interprétation, qui utilise également des informations sur le contexte de l'écriture. En ce qui concerne le contexte linguistique, il est utilisé aux deux derniers niveaux de la reconnaissance, à savoir celui des mots et enfin celui des phrases.

La reconnaissance de mots isolés utilise généralement des dictionnaires pour pouvoir lever des ambiguïtés sur la reconnaissance des caractères, en ne considérant comme réponse que les mots appartenant au dictionnaire. L'utilisation de ces informations linguistiques se situe donc au niveau lexical.

Il est aussi intéressant de tirer partie des connaissances linguistiques au niveau syntaxique. Dans ce cas, les informations concernent le contexte du mot écrit et notamment les mots qui l'entourent. Ces connaissances linguistiques sont généralement représentées grâce à des modèles de langage : ils permettent de définir les suites de mots possibles d'une langue donnée.

La figure I.3 donne un exemple de l'apport des contextes linguistiques. [2]

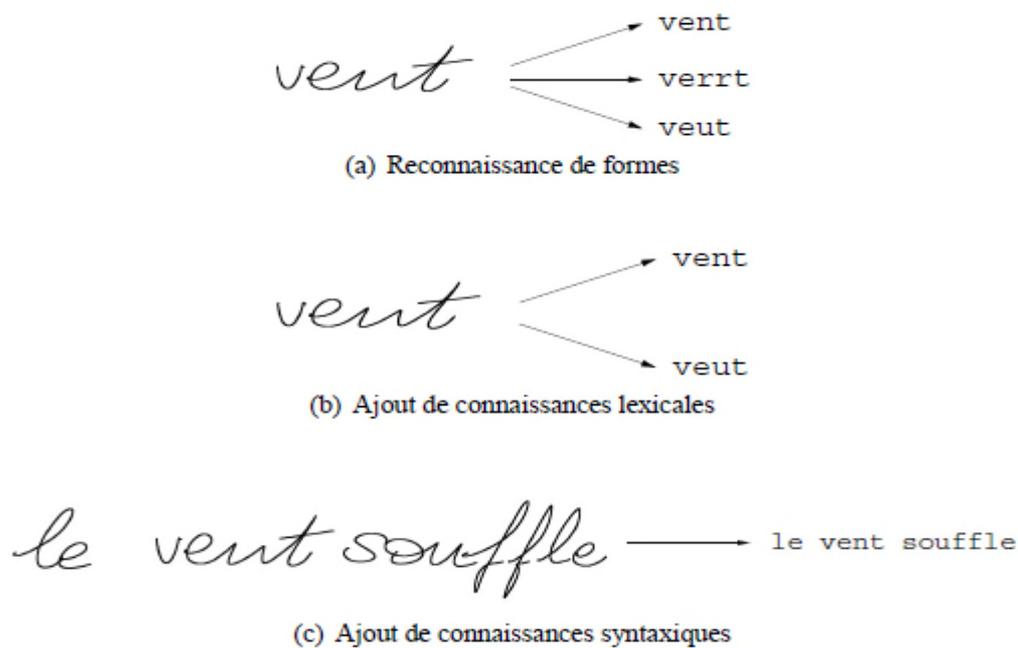


Figure I-3: Importance de la prise en compte du contexte des mots.

Dans la figure I.3(a), seule la forme du mot est prise en compte. Trois mots peuvent alors être proposés, comme résultat : vent, verrt et veut. En prenant en compte le contexte lexical (voir figure I.3(b)), grâce à un dictionnaire de la langue française, le « mot » verrt peut être écarté puisqu'il ne s'agit pas d'un mot correct, dans la langue française. Enfin lorsque le contexte dans lequel a été écrit le mot est pris en compte (voir figure I.3(c)), comme ici la phrase entière, il est possible de désambiguïser le résultat de reconnaissance, en ne conservant que le mot vent.[2]

Il existe deux représentations possibles du signal manuscrit: la représentation hors-ligne ou la représentation en-ligne.

I.2 PROBLEMATIQUE EN-LIGNE \ HORS-LIGNE:

La reconnaissance de l'écriture manuscrite est actuellement un domaine très actif où l'on distingue communément deux secteurs d'application en reconnaissance de l'écriture manuscrite suivant le mode d'acquisition ou de saisie de l'écriture:

- la reconnaissance dite dynamique ou temps-réel, ou encore en-ligne. Le texte est reconnu par l'ordinateur à partir de la trajectoire du stylo décrite par une suite de points.
- la reconnaissance dite statique ou encore hors-ligne. Le texte a été écrit sur du papier et doit être reconnu par l'ordinateur à partir de son image.

1.2.1 L'ECRITURE EN LIGNE:

L'utilisateur écrit naturellement à l'aide d'un stylet sur une ardoise ou un écran ou d'un stylo digital. Le logiciel de la reconnaissance interprète les caractères ou les mots écrits pour les transformer en caractères numériques.

Trois propriétés caractérisent la reconnaissance en ligne: la notion d'ordre d'écriture (enchaînement temporel de traits), la dynamique du tracé(vitesse, accélération, lever du stylo) et squelette du tracé(aucune épaisseur de trait).



Figure I-4: Systèmes d'acquisition d'écriture en ligne.

La figure I.4 présente différents outils et supports d'acquisition [3]. Les premiers logiciels de reconnaissance manuscrite commerciales ont été intégrés a des organisateurs électroniques muni d'un stylet permettant la saisie de caractères voire de mots et phrases aujourd'hui et parfois muni de clavier.En dehors de ce marché important des petits assistants personnels, d'autres applications se sont développées a partir de tablette graphique et récemment des stylos digitaux:

Dans le milieu médical, pour la saisie et stockage d'information auprès du lit des malades, pour la saisie d'ordonnance médicale.

Dans le monde de l'éducation, pour seconder l'enseignement dans sa tache d'apprentissage de l'écriture qui ne peut surveiller qu'un enfant a la fois pendant son geste

de production d'écriture. Autant dans le milieu scolaire que médical, pour détecter rapidement les différentes causes liées aux troubles psychiques et moteurs (parkinson, sclérose...) et aux échecs scolaires (dyslexie).

Et dans l'univers des réunions, avec la possibilité de prise de notes, annotations, conservations de toutes traces écrites et orales des différents intervenants de la réunion.

Les principaux axes de recherche autour de la reconnaissance en-ligne peuvent être résumés comme suit:

- la reconnaissance de mots, de phrases, de textes en utilisant des connaissances contextuelles (spécifiques au document, linguistiques, etc.) ;
- l'adaptation automatique à l'écriture d'un scripteur à partir d'un système de reconnaissance générique ;
- la présentation des résultats de reconnaissance, l'ergonomie de l'interface stylo, l'édition d'un document ;
- les outils d'éducation d'aide à l'apprentissage de l'écriture, la détection de troubles liés à l'écriture ;
- l'authentification du scripteur, la reconnaissance de signatures.

1.2.2 L'ECRITURE HORS-LIGNE:

L'autre cadre applicatif de la reconnaissance de l'écriture manuscrite est la reconnaissance "hors-ligne", la reconnaissance a lieu après l'étape de numérisation du document papier et n'a pas forcément un objectif de traitement temps réels comme en reconnaissance en-ligne. Une difficulté importante est d'isoler l'écrit du reste du document qui peut être bruité par le système de numérisation mais aussi par son contexte. Le logiciel doit dans l'image bidimensionnelle séparer les mots avant de les reconnaître. Les difficultés sont loin d'être entièrement résolues et les logiciels qui fonctionnent actuellement sont encore très ciblés sur des applications précises telles que:

- La lecture automatique de documents manuscrits: lecture d'adresse postale manuscrite et tri automatique du courrier, lecture des montants de chèques.
- La reconnaissance et traitement automatique d'information manuscrites
- pré casées: analyse de formulaires pour les grandes administrations principalement.
- La numérisation des collections patrimoniales, des archives, documents anciens.
- La recherche d'information dans une base de documents manuscrits telle que l'identification du scripteur.

Ces logiciels de reconnaissance ont un coût très lourd en terme de puissance de calcul, mémoire et financier et ne s'appliquent pas aux particuliers ou petites infrastructures contrairement aux logiciels de reconnaissance en-ligne destinés davantage à des applications légères et à des ordinateurs, agendas de poche.

La figure I.5 illustre ces deux types de représentation du signal manuscrit. Dans un document hors-ligne (voir figure I.5(a)), le texte est représenté par une image, obtenue par numérisation du document original. Le signal d'entrée du système de reconnaissance est alors une matrice de pixels en niveau de gris. Pour la reconnaissance, il faut au préalable identifier les parties de l'image qui correspondent aux zones d'écriture.

Un document en-ligne (voir figure I.5(b)) est produit en utilisant un stylet sur un dispositif électronique (tablette, assistant personnel, par exemple). Le signal consiste en une suite de points, ordonnés temporellement. Les informations dynamiques de ce signal en font une représentation plus riche que le signal hors-ligne. Cela permet de pouvoir connaître la vitesse ou l'accélération, par exemple, mais surtout le sens et l'ordre du tracé. [2]

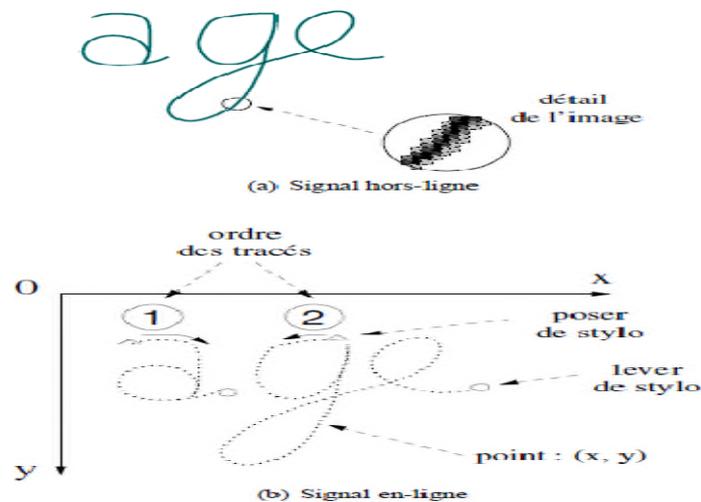


Figure I-5: Représentations du signal du mot "age".

Bien que le cadre de cette étude soit la reconnaissance d'écriture manuscrite en-ligne, cet état de l'art recouvre également des travaux sur la reconnaissance hors-ligne. En effet, plus la reconnaissance se trouve à un haut niveau (lettre puis mots puis textes), moins il existe de différences entre les approches de reconnaissance hors-ligne et en-ligne. Les domaines applicatifs sont cependant différents. Les approches utilisées pour l'intégration de connaissances linguistiques seront alors quasiment indépendantes de la représentation

du signal manuscrit. De même, nous pourrions nous inspirer des approches de segmentation hors-ligne, malgré les différences entre les approches de segmentation, car les recherches dans le domaine de la reconnaissance hors-ligne sont généralement plus avancées que celles en reconnaissance en-ligne.

I.2.3 RECONNAISSANCE:

La principale différence entre les deux domaines de l'écriture manuscrite en ligne et hors-ligne, souvent traités séparément, réside dans la nature des données à traiter (temporelles ou spatiales) et de l'information pertinente que l'on peut extraire dans un but de reconnaissance. Les schémas de reconnaissance sont pour autant globalement communs et se déclinent en trois notions clés: les prétraitements, la reconnaissance et les post-traitements.

- **Les prétraitements** concernent l'acquisition et la normalisation des données. Ils servent à supprimer les bruits induits par le contexte lui-même lors de l'acquisition (fréquence d'échantillonnage de la tablette graphique, qualité de numérisation du scanner, extraction du texte dans un document) et ceux générés par l'humain. Dans une application mono utilisateur, on cherchera à rendre les données indépendantes des variabilités de l'utilisateur pendant sa production d'écriture: vitesse, forme. Dans une application omni scripteur, le système devra être à la fois indépendant des variabilités inter et intra scripteur.
- **La reconnaissance** peut être décomposée en deux parties: la segmentation et la reconnaissance propre des données. La partie segmentation consiste à découper les données en sous-éléments comme les lettres dans un mot, ou des mots dans un texte qui seront alors identifiés et étiquetés. (Dans notre étude, nous n'aurons pas besoin de la segmentation, étant donné que nous travaillerons sur la reconnaissance de caractères isolés).
- **Extraction de caractéristiques:** Dans cette section, nous présentons un aperçu des méthodes d'extraction de caractéristiques utilisées en reconnaissance de caractères manuscrits. En réalité, en plus de la taxonomie présentée dans la figure I.6, le domaine de la reconnaissance de caractères peut être décrit par la méthode de collection de données, des méthodes d'extraction de caractéristiques, des méthodes de classification ou du format de représentation des données.

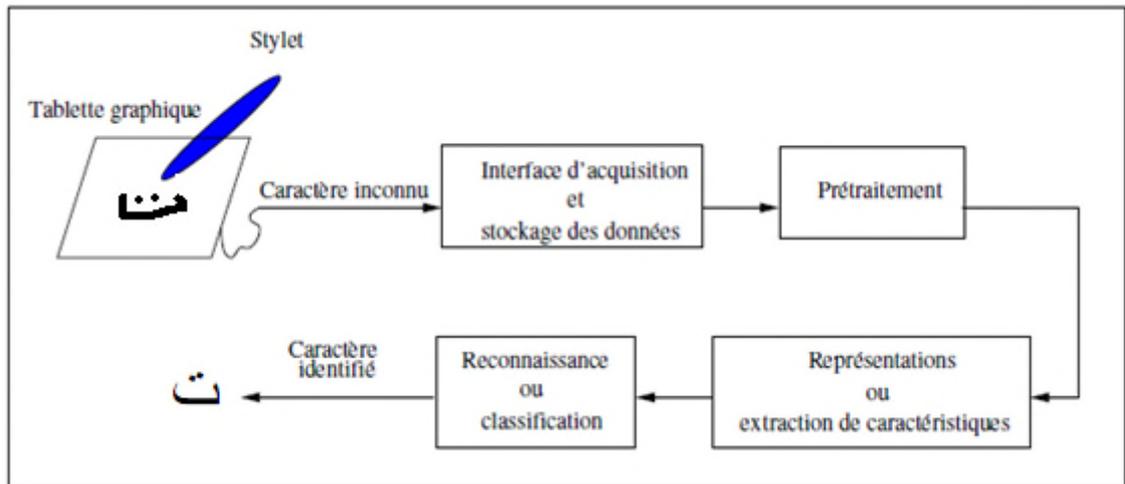


Figure I-6: Schéma général d'un système de reconnaissance d'écriture manuscrite.

- **Les post-traitements** permettent de simplifier ou corriger l'étape précédente par l'apport de connaissances lexicales, grammaticales et sémantiques. Les connaissances lexicales sont généralement les plus utilisées dans tout le système de reconnaissance d'écriture; ils permettent de vérifier la présence d'un mot dans un dictionnaire, cette vérification peut s'avérer coûteuse en fonction de la taille du lexique choisi. De nombreux travaux ont porté sur l'optimisation de cette opération: accès au lexique, architecture du lexique, réduction du lexique. L'introduction de modèles de langages permet significativement d'améliorer les taux de reconnaissance de l'écriture manuscrite. Cet axe est relativement récent dans le domaine de l'écrit et se base sur des travaux antérieurs en reconnaissance de la parole.

I.2.4 VERIFICATION DES SIGNATURES

La vérification de signatures, comme son nom l'indique, est facile à comprendre, par exemple la carte bancaire à puce avec le code PIN est le mode de paiement privilégié. La signature manuscrite est la seule façon d'authentifier le détenteur de la carte bancaire. Pour l'instant, la vérification de la signature est simplement une comparaison de deux images, l'une sur la carte et l'autre sur la facture, par le commerçant, ce qui n'est souvent pas très efficace. De même, les documents administratifs sont souvent validés par une signature manuscrite. Bien qu'un certain nombre de systèmes de vérification de signature aient déjà été proposés, il n'existe pas à ce jour de mise en œuvre systématique et à grande échelle de système de vérification automatique de signature manuscrite. Ceci dit, la vérification d'identité par la signature manuscrite devient très importante, vu

l'utilisation répandue et la grande acceptabilité de cette modalité.

- **Un geste compliqué et unique**

Presque chaque personne qui sait écrire, a sa propre signature manuscrite. On a l'habitude de signer sur un grand nombre de documents qui sont peut-être simplement une lettre, une facture ou un contrat... Ces documents deviennent valides avec une simple image de notre signature. Dans quelques cas, on compare cette image avec celle à disposition pour être sûr de l'authenticité de la signature. Malheureusement, notre signature n'est pas secrète ce qui permet à un faussaire potentiel de s'entraîner pour reproduire une imitation. Si l'imitateur s'entraîne suffisamment bien, il arrive à reproduire des images assez similaires aux images originales, ce qui pourra tromper un examinateur. Dans ce cas, on s'intéresse seulement à l'image de la signature, dénommée la signature hors-ligne, mais pas à la façon dont elle a été produite.

Au contraire, la signature en-ligne contient non seulement l'image mais aussi les informations dynamiques du geste de signer. En fonction du type de capteur, soit la trajectoire de la signature est seule échantillonnée et sauvegardée, sinon, d'autres informations sont aussi enregistrées comme la pression, les angles d'inclinaison du stylo.

Le geste de signer est différent pour chaque personne. Il est aussi difficilement falsifiable car il n'est pas possible de récupérer les informations dynamiques du geste à partir de l'image seule d'une signature. Avec cette information en plus de l'image de la signature, la vérification de la signature devient un outil plus efficace de vérification d'identité.

La différence entre la signature en-ligne et la signature hors-ligne est liée à la phase de capture et à la nature des données stockées dans la mémoire. La signature hors-ligne, comme celle que l'on voit sur le papier, est simplement une image, numérisée de façon différée à l'aide d'un scanner ou d'une caméra numérique. La signature en-ligne est numérisée directement par un dispositif qui permet d'échantillonner les signaux à une fréquence fixe, au moment où l'on signe. En conséquence, l'acquisition d'une signature en-ligne nécessite un capteur spécifique. Une tablette à digitaliser ou un écran tactile d'un PDA suffisent pour cette tâche.

Durant ces deux dernières décennies, il y a eu plusieurs développements dans le domaine de vérification des signatures manuscrites. Toutefois, le problème des imitations reste

toujours ouvert. En effet, afin de décider de l'authenticité d'une signature, les services judiciaires, ainsi que les banques font toujours appel à l'expertise humaine, d'où le besoin de développer un système objectif capable d'identifier les signatures falsifiées. En effet, le contrôle des falsifications par un système hors ligne possède un rôle important dans différents domaines d'applications tels que la médecine (les prescriptions, les rapports médicaux...), le commerce (chèques bancaires, contrats...), les domaines politique et juridique...

La littérature montre que pour l'être humain, la différenciation entre les signatures authentiques et leurs faux simples ou aléatoires est une tâche relativement aisée.

En effet, un simple coup d'œil permet de juger de l'authenticité d'un faux aléatoire ou encore d'un faux simple puisque ce type de faux présente une nette dissemblance morphologique par rapport à la signature authentique. Cependant, la tâche n'est pas triviale dans le cas des faux par imitation notamment après une phase préalable d'apprentissage

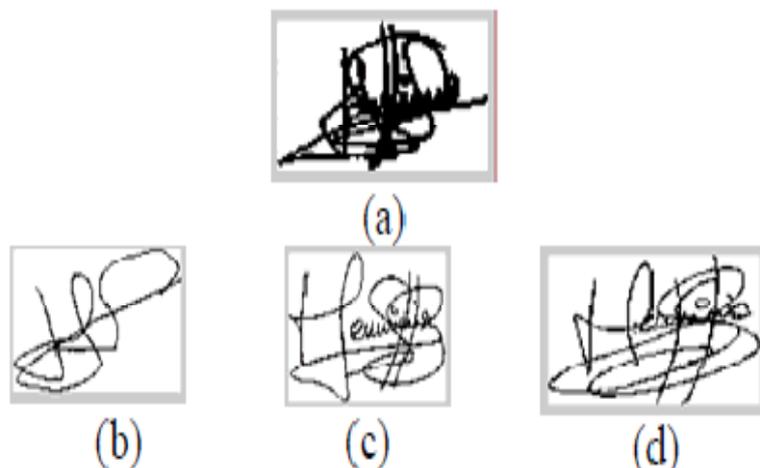


Figure I-7: Une signature authentique (a) avec des échantillons imités (b, c et d) extraits à partir de la base de test.

I.2.5 QUELQUES EXEMPLES D'APPLICATIONS

Voici quelques exemples d'applications informatiques utilisant des systèmes de reconnaissance d'écriture :

- 1- prendre des notes avec un stylo électronique directement sur l'écran de son PDA ou tablet PC et pouvoir les éditer avec un logiciel de traitement de texte.

- 2- rédiger des messages ou des mails à l'aide d'un stylo sur l'écran de son Smartphone.
- 3- saisir des formules mathématiques sur l'écran de son tablet PC.
- 4- rétroconvertir des documents papiers : par exemple pour numériser à l'aide d'un scanner un document manuscrit et le transformer en document éditable avec un logiciel de traitement de texte.
- 5- indexer des documents d'archives : ce sont souvent des documents abîmés et peu accessibles au public, les numériser et reconnaître certaines parties du document (par exemple le nom sur une fiche d'état civil) permet de faciliter les recherches.
- 6- trier automatiquement le courrier en analysant automatiquement des adresses postales sur des enveloppes.
- 7- traiter des formulaires remplis à la main : comme par exemple un sondage, un bon de commande ou un chèque.

Ces différents exemples correspondent à l'analyse de documents plus ou moins complexes contenant de l'écriture manuscrite. Suivant les cas, la reconnaissance consiste à reconnaître des documents complets, des mots ou seulement des caractères dans une partie des documents.

I.2.6 LES DIFFERENTS TYPES DE DOCUMENTS

Les trois premiers exemples illustrent la reconnaissance de documents en-ligne : les documents de ce type sont constitués d'une suite temporelle de points correspondant au passage du stylo électronique sur l'écran. L'emplacement de l'écriture dans le document est donc connu ainsi que l'ordre des différents tracés. Les quatre derniers exemples correspondent à de la reconnaissance de documents manuscrits hors-ligne. Un scanner ou une caméra permet de transformer chaque document papier en image, et c'est cette image qui est ensuite traitée et dans laquelle l'écriture est reconnue. Reconnaître de l'écriture dans des documents hors-ligne est souvent une tâche plus difficile car :

- ces documents peuvent être abîmés ou mal numérisés ;
- il faut retrouver dans l'image les zones contenant de l'écriture et l'ordre des tracés n'est pas connu.

Les systèmes de reconnaissance peuvent être :

- omni-scripteurs : le système est très général et est sensé reconnaître tout type d'écriture (pour mettre en valeur ce point, de tels systèmes sont évalués sur des bases d'écritures decorrélées de celles de l'apprentissage) ;
- multi-scripteurs : le système est capable de reconnaître les écritures de plusieurs scripteurs (les bases d'apprentissage et de tests sont différentes mais écrites par les mêmes personnes) ;
- mono-scripteurs : le système est spécialisé pour l'écriture d'un scripteur.

I.3 PRINCIPES GENERAUX SUR L'ECRITURE EN-LIGNE

Contrairement aux documents papier qui sont numérisés sous forme d'images, les documents saisis en-ligne (et plus spécifiquement les caractères et gestes) sont stockés sous forme d'encre électronique. Les documents en-ligne peuvent être saisis en utilisant plusieurs types de périphériques :

- la souris d'un ordinateur de bureau ;
- une tablette graphique sans retour visuel ;
- un écran tactile sur lequel on dessine avec le doigt ou un crayon passif (guichet automatique, PDA, smartphone...) ;
- un écran sensitif utilisé avec un stylo spécifique (TabletPC...) ;
- un stylo enregistrant sa position absolue (comme le stylo Anoto utilisant une caméra sur un papier spécifique) ;

Ces différents périphériques font que l'encre électronique enregistrée pourra être de nature et de qualité différentes. Les domaines applicatifs sont tout aussi vastes, de l'écriture de textes à la saisie de schémas en passant par le remplissage de formulaires ou les gestes d'édition de documents.

Mais la principale difficulté rencontrée lors de la reconnaissance de l'écriture manuscrite est la variabilité des styles d'écriture. En effet, la forme des caractères manuscrits varie énormément d'un scripteur à l'autre et même pour un scripteur donné suivant le contexte du caractère (sa position dans le mot, les lettres voisines...). Cette variabilité est même source d'ambiguïté entre les caractères puisque un même tracé peut avoir différentes significations suivant le contexte ou selon le scripteur. Ces propriétés font que la reconnaissance de l'écriture manuscrite est un domaine applicatif pour la reconnaissance de formes très riche en difficultés et en défis.

Dans ces travaux nous nous limitons à la reconnaissance de caractères isolés (lettres ou

chiffres). En effet cette problématique, déjà très étudiée, est la brique de base de beaucoup de systèmes plus complexes permettant la reconnaissance de mots, de phrases, de textes. Une petite amélioration de l'efficacité de la reconnaissance des caractères peut permettre de diminuer grandement la complexité des étapes suivantes. Le fait que nous nous trouvions dans un contexte de reconnaissance embarquée ajoute des contraintes sur le système de reconnaissance dues aux ressources limitées des périphériques mobiles.

Ces contraintes limitent les stratégies de reconnaissance des formes que nous pouvons utiliser. Dans la suite, nous présentons quelques notions de reconnaissance des formes en mettant en valeur les points importants pour la compréhension de la suite de ce manuscrit.

I.3.1 FORMALISATION DE LA RECONNAISSANCE DE FORMES

Le problème de reconnaissance des formes peut être formalisé par une fonction de classement qui à une entrée e décrite dans l'espace E associe une sortie s d'un espace S :

$$f : E \rightarrow S$$

$$\forall e \in E, \exists s \in S : f(e) = s$$

Les entrées du système représentent la forme à reconnaître et les sorties les classes auxquelles elles appartiennent. La fonction de décision f est concrétisée par un classifieur. Nous présentons donc maintenant les différents types d'espaces d'entrée utilisés en reconnaissance de l'écriture manuscrite en-ligne, les différentes natures de sorties possibles et les deux modélisations principales pour représenter les connaissances du classifieur.

I.3.1.1 Les entrées du classifieur

D'une façon générale, l'entrée d'un classifieur est une forme qui correspond à une observation de l'environnement extérieur au système. Cette forme est acquise au moyen d'un capteur et présentée sous la forme d'un signal (temporel ou statique). Dans notre contexte, ce signal brut est appelé encre électronique. Il peut être présenté directement au classifieur ou bien peut subir au préalable un ensemble de modifications allant du simple pré-traitement jusqu'à l'extraction de caractéristiques.

I.3.1.1.1 L'encre électronique

L'encre électronique décrit un tracé manuscrit acquis via un périphérique enregistrant différents paramètres au court du temps. Le contenu de ce signal temporel dépend donc

du matériel utilisé. D'une façon générale, l'encre électronique est définie par une fonction paramétrique $p(t)$ discrète donnant la position du crayon et diverses informations complémentaires. Il s'agit donc d'un vecteur d'au moins deux dimensions pouvant représenter à l'instant t :

- les coordonnées du crayon, notées $x(t)$ et $y(t)$, dans un repère absolue ;
- la pression du crayon sur la surface, nulle (ou négative) si le crayon ne touche pas la surface ;
- l'inclinaison du crayon par rapport à la surface (spécifiée par deux angles) ;
- un mode du tracé (certains crayons possèdent un bouton ou une gomme) ;
- ...

Suivant le type d'application, différentes informations peuvent être sélectionnées. D'un autre côté, de nouvelles informations peuvent être calculées à partir de celles-ci comme la vitesse instantanée ou la courbure du tracé. Le temps écoulé entre deux points est généralement constant et détermine le taux d'échantillonnage du signal qui dépend du type de périphérique utilisé. Plus ce taux est élevé plus l'encre électronique est fidèle au tracé réellement effectué par l'utilisateur. Ce signal brut peut être utilisé directement comme entrée du classifieur. Mais ce mode de fonctionnement peut poser plusieurs problèmes. Il impose notamment de pouvoir traiter des signaux dans lesquels la quantité d'information utile peut varier (en fonction de la taille de la forme, de la durée d'acquisition, de la précision du capteur). C'est pourquoi ce signal brut subit très souvent une série de pré-traitements.

L'objectif de ceux-ci est d'obtenir une description de la forme qui soit la plus stable possible, c'est-à-dire qu'entre différentes acquisitions de la même forme, les signaux obtenus doivent être les plus semblables possibles. Ces traitements dépendent fortement du cadre applicatif.

L'encre électronique peut subir des traitements cherchant à résumer l'information, à normaliser le signal ou supprimer des défauts de l'acquisition : la segmentation isole les entités indépendantes dans le signal (trouver les caractères dans un mot par exemple) ;

- le ré-échantillonnage temporel ou spatial permet de palier des problèmes d'acquisition imparfaite (point manquant, intervalle de temps non régulier) ou de réduire la quantité d'information (i.e. le nombre de points) ;
- les transformations géométriques normalisent certaines variabilités des styles d'écriture (taille, inclinaison, rotation...) ;

- le filtrage permet de supprimer les points aberrants dus à des problèmes d'acquisition ;
- ...

Nous utilisons du signal d'origine seulement les points du tracé qui ont une pression strictement positive, c'est-à-dire les points où le crayon touche la surface sensible. Pour certains caractères écrits en plusieurs traits, il y a donc une discontinuité dans le signal. Chaque trait est séparé par un point où la pression est nulle et est appelé trace. Un tracé est donc composé d'une ou plusieurs traces. Ensuite seuls les coordonnées des points sont utilisés pour définir l'encre électronique, nous avons donc :

$$p(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}$$

Mais ce signal n'est pas utilisé directement en entrée du classifieur, en effet un certain nombre de caractéristiques sont extraites de l'encre électronique.

1.3.1.1.2 Les caractéristiques utilisées

Pour réduire la complexité du signal d'origine, une possibilité très souvent utilisée consiste à extraire du signal brut un certain nombre de caractéristiques ou attributs décrivant la forme. Celles-ci sont très souvent numériques (quantité de points, centre de gravité, position des points remarquables, ...) mais peuvent aussi être symboliques.

L'objectif est d'obtenir une description à la fois réduite et la plus pertinente possible pour pouvoir différencier les classes et donc faciliter le travail du classifieur. L'utilisation de caractéristiques en entrée du classifieur simplifie donc la tâche de classification mais nécessite une sélection préalable des caractéristiques pour bien les choisir et un traitement supplémentaire de l'encre pour les extraire pendant l'utilisation.

Dans la suite nous ne considérerons que le cas général de la représentation des entrées par un vecteur $e = [e_1, \dots, e_n]$ de l'espace E à n dimensions. Chaque dimension représente une caractéristique particulière ayant son propre espace de définition et prenant des valeurs numériques. C'est sur cet espace E qu'il s'agit maintenant de définir la modélisation du classifieur.

1.3.1.2 Modélisation intrinsèque et modélisation discriminante

Une des principales difficultés en reconnaissance de formes est de choisir et de concevoir le classifieur et la modélisation sur laquelle il repose. Nous définissons la modélisation

d'un classifieur comme le choix des connaissances utilisées dans le classifieur et surtout la façon dont elles sont organisées. Les connaissances d'un classifieur sont les éléments du système de reconnaissance sur lesquels reposent la modélisation et qui permettent la classification. Elles sont soit issues de connaissances a priori injectées par un expert, soit issues d'une extraction automatique à partir de données.

En général deux types de modélisation peuvent être distingués selon leur objectif :

- modélisation discriminante : description implicite des classes par la définition des frontières de décision pour les discriminer ;
- modélisation intrinsèque : description explicite des classes par la définition de leurs propriétés intrinsèques.

La figure I.8 schématise la différence entre ces deux modélisations pour un exemple à trois classes définies dans deux dimensions.[1]

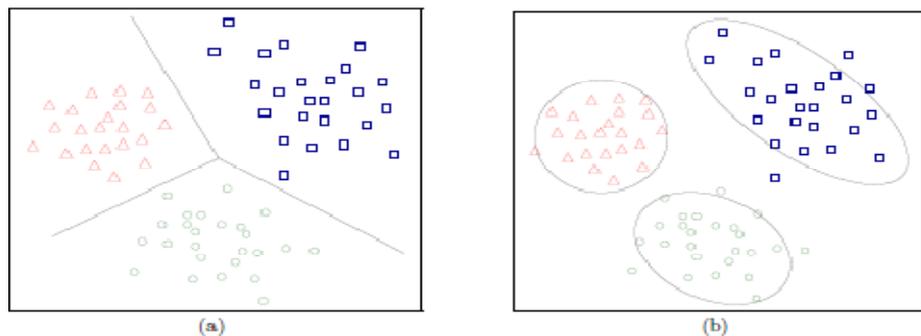


Figure I-8: Deux types de modélisation : (a) discriminante et (b) intrinsèque

Dans la modélisation discriminante, le but est de chercher, dans l'espace E de représentation des formes, les frontières séparant le mieux les classes (une hypersurface dans le cas général). Les classes sont donc comparées les unes aux autres afin de rechercher ce qui les différencie pour déterminer les frontières de décision. Pour la classification, il suffit de trouver de quel côté de la frontière se trouve la forme à classer. Cette modélisation très proche de l'objectif de classification est souvent très performante car seules les connaissances utiles pour la classification sont considérées.

Dans la modélisation intrinsèque, l'objectif n'est pas de trouver directement les frontières séparant les classes mais plutôt de les décrire par des propriétés les caractérisant dans l'espace de représentation. On parle aussi de modélisation générative puisqu'à partir de cette représentation, il est possible de générer des formes dans E fidèlement à la

répartition de chaque classe. Cette caractérisation prend souvent la forme d'un prototype servant de modèle pour chaque classe. Ce sont ces prototypes qui servent de connaissance au classifieur. Cette modélisation est souvent utilisée pour mesurer la similarité entre une forme en entrée et les différentes classes modélisées. Cette modélisation rend un peu plus interprétable le système grâce aux prototypes. Pour effectuer la classification, il suffit de mettre en concurrence les modèles de chaque classe par le biais d'une fonction de décision plus ou moins simple.

L3.1.3 Les sorties du classifieur

La sortie du classifieur doit contenir assez d'information pour désigner la classe de la forme reconnue. L'ensemble des classes possibles est généralement connu a priori par expertise du problème et noté $\{c_1, \dots, c_k, \dots, c_C\}$ pour C classes. Nous ne traiterons pas le cas où cet ensemble n'est pas initialement connu.

Quelque soit la modélisation choisie pour le classifieur, plusieurs natures d'espaces de sortie S peuvent être utilisées en fonction de l'information supplémentaire qu'elles apportent en plus de la classification. Quatre types de sorties sont généralement distingués:

- le type classe : une seule classe est fournie en sortie, celle jugée la plus adaptée à l'entrée présentée ;
- le type ensemble : un ensemble de classes possibles est rendu sans priorité, peu utilisé car difficilement exploitable en pratique ;
- le type rang : extension du type précédent, un ordre de préférence est donné pour toutes les classes (ou parfois un sous-ensemble) ;
- le type mesure : extension du type précédent, les classes sont ordonnées suivant une quantité (mesure de confiance, score, probabilité...).

Dans la suite, nous considérons le cas général où la sortie du classifieur est un vecteur $[s_1, \dots, s_k, \dots, s_C]$ S où chaque élément s_k est un indicateur sur l'adéquation de la forme présentée en entrée par rapport à la classe c_k . Parmi les nombreuses possibilités de mesure d'adéquation, nous allons dégager trois types de scores qui correspondent à trois qualités qui ne sont pas mutuellement exclusives :

- Les scores probabilistes : les sorties du classifieur donnent directement la probabilité $P(k|X)$ de la classe k sachant la forme X , ou bien ces probabilités sont calculées à partir des $P(X|k)$ grâce à la relation de Bayes :

$$p(k \setminus X) = \frac{p(k \setminus X)p(k)}{\sum_j p(k \setminus j)p(j)} = \frac{p(k \setminus X)p(k)}{p(X)} \quad (I.1)$$

dans ce cas, la somme des s_k est toujours 1 ;

- les scores discriminants : le score informe juste sur la décision, i.e. de quel côté de la frontière est la forme (par exemple score positif ou négatif pour un SVM) ;
- les scores intrinsèques : chaque score s_k donne un degré d'appartenance de la forme à la classe correspondante c_k , les scores sont indépendants entre eux, donc plusieurs peuvent être fortement activés ou au contraire aucun.

I.4 Conclusion

Ce mémoire se place dans le cadre de la reconnaissance d'écriture manuscrite et plus particulièrement dans celui de la reconnaissance de caractères isolés, saisis en-ligne sur des périphériques tels que des Tablet PC.

La principale différence entre les deux domaines de l'écriture manuscrite en ligne et hors-ligne, souvent traités séparément, réside dans la nature des données à traiter (temporelles ou spatiales) et de l'information pertinente que l'on peut extraire dans un but de reconnaissance.

CHAPITRE II: RECONNAISSANCE DE L'ECRITURE ARABE MANUSCRITE

II.1 PROBLEMATIQUE

II.1.1 PRESENTATION DE LA LANGUE ARABE

L'arabe littéral, arabe moderne unifiée ou encore classique est le nom que l'on donne à une variante de la langue arabe, utilisée comme langue officielle dans tous les pays arabes, et comme langue commune entre pays arabes. Elle est également employée dans la plupart des écrits et, à l'oral, dans les situations officielles ou formelles (discours religieux, politiques, journaux télévisés).

L'arabe littéral se distingue ainsi de l'arabe dialectal, qui est la langue vernaculaire parlée au quotidien et ce depuis l'expansion de l'islam. Cette variété de la langue recouvre plusieurs dialectes locaux pouvant varier assez fortement d'un pays à l'autre. Dans tous les pays arabes, un dialecte national composé par plusieurs dialectes locaux est parlé. Aucun d'entre ces dialectes n'est identique complètement à l'arabe classique ou littéraire.

Le monde arabe se compose de vingt-deux pays s'étendant de la Mauritanie à l'ouest, au sultanat d'Oman à l'est. Sa population est estimée à 355 millions de personnes. Il se divise en Machreq à l'est, et en Maghreb à l'ouest (la frontière entre le Machreq et le Maghreb est le Nil). L'intercompréhension est possible, voire facile, entre les locuteurs des variantes d'arabe dialectal du Machreq, de même l'est-elle entre les locuteurs des variantes d'arabe dialectal d'Afrique du nord, mais elle est plus difficile entre les locuteurs du Machreq et du Maghreb.

Dans la réalité des échanges linguistiques, il n'y a pas de séparation étanche entre arabe littéral et arabe dialectal, mais plutôt un continuum où dominent les formes mixtes. Néanmoins, les spécificités géographiques et historiques influencent les différentes versions de l'arabe dialectal. Citons par exemple :

- l'influence du berbère pour les dialectes maghrébins
- l'influence de l'égyptien ancien pour l'égyptien

- l'influence du phénicien pour le syro-libano-palestinien, le tunisien, l'algérien et le maltais
- l'influence de l'italique pour le maltais
- les emprunts français, italiens et castillans pour les dialectes maghrébins et le libanais
- les emprunts turcs pour le syro-libano-palestinien.

En ce qui concerne les styles d'écriture, les différences sont également importantes. Ainsi, la plupart des systèmes présentés lors de la compétition ICDAR 2007 [4], entraînées sur des données tunisiennes, accusent une baisse de performances de 10 à 15 points entre les résultats obtenus sur un ensemble de test en provenance de Tunisie, et un autre ensemble de test en provenance des Émirats-Arabes-Unis. La proportion de ligatures verticales est plus élevée chez les scribes du Moyen-Orient, et la variabilité des styles d'écriture est également plus importante. Des adaptations de la chaîne et des réentraînements semblent nécessaires. Notons également que le farsi (persan), utilisé principalement en Iran et en Afghanistan, partage un grand nombre de points communs avec l'écriture arabe. Cet aspect est hors du cadre de cette thèse : nous nous restreignons ici à l'étude de l'écriture arabe manuscrite sur des données en provenance de scribes tunisiens. Sauf mention contraire, les données utilisées proviennent de la base IFN/ENIT [5], une base de noms de villes tunisiennes.

Tableau II-1: Les lettres arabes[6]

lettre	nom	fin	milieu	début	lettre	nom	fin	milieu	début
ا	alif	آ	أ	إ	ط	Ta	ظ	ظ	ظ
ب	ba	ب	ب	ب	ظ	tha	ظ	ظ	ظ
ت	ta	ت	ت	ت	ع	3ayn	ع	ع	ع
ث	tha	ث	ث	ث	غ	ghayn	غ	غ	غ
ج	jim	ج	ج	ج	ف	fa	ف	ف	ف
ح	ha	ح	ح	ح	ق	qaf	ق	ق	ق
خ	kha	خ	خ	خ	ك	kaf	ك	ك	ك
د	dal	د	د	د	ل	lam	ل	ل	ل
ذ	dhal	ذ	ذ	ذ	م	mim	م	م	م

ر	ra	ر	ر-	ر-	ن	nun	ن	ن	ن
ز	zay	ز	ز-	ز-	ه	ha	ه	ه	ه
س	sin	س	س-	س-	و	waw	و	و-	و-
ش	shin	ش	ش-	ش-	ي	ya	ي	ي	ي
ص	sad	ص	ص-	ص-	ء	hamza	أ و إ ئ		
ض	Dhad	ض	ض-	ض-					

II.1.2 CARACTERISTIQUES MORPHOLOGIQUES DE L'ECRITURE ARABE

L'écriture arabe est semi-cursive aussi bien dans sa forme imprimée que manuscrite. Les caractères d'une même chaîne (ou pseudo-mot) sont ligaturés horizontalement et parfois verticalement (dans certaines fontes deux, trois et même quatre caractères peuvent être ligaturés verticalement), occultant ainsi toute tentative de segmentation en caractères. De plus, la forme d'un caractère diffère selon sa position dans le pseudo-mot et même dans certains cas, selon le contexte phonétique. En outre, plus de la moitié des caractères arabes incluent dans leur forme des points diacritiques (1,2 ou 3). Ces points peuvent se situer au-dessus ou au-dessous du caractère, mais jamais en haut et en bas simultanément. Plusieurs caractères peuvent avoir le même corps mais un nombre et /ou une position de points diacritiques différents. D'autre part, le caractère arabe présente une forme cursive voyellée nécessitant, pour la majorité des lettres, des matrices de dimensions importantes. Ceci laisse jusqu'à présent les formes informatisées des caractères arabes non encore normalisées.

Le mot arabe n'a pas de longueur fixe, il peut comprendre un ou plusieurs pseudo-mots incluant chacun un nombre souvent différent de caractères. L'étude de la morphologie des pseudo-mots montre que l'écriture arabe présente des variations dans des bandes horizontales plus ou moins complexes en fonction de la calligraphie des caractères contenus dans le pseudo-mot [7]. La bande centrale est généralement la plus chargée au point de vue densité d'informations en pixels. Elle correspond aux lieux des ligatures horizontales, aux caractères centrés (sans extensions), aux boucles... .

La Figure II.1 donne des exemples d'écritures arabes imprimées et manuscrites [9]. Pour plus d'informations sur les caractéristiques morphologiques de l'arabe, nous renvoyons le lecteur aux références suivantes : [7,8].

Par ailleurs, les différents travaux élaborés en AOCR, montrent que la cursivité de l'écriture arabe, la complexité de la morphologie des caractères, les élongations des ligatures horizontales ainsi que les combinaisons verticales de certains caractères, constituent les problèmes majeurs liés au traitement de cette écriture. En effet, ces problèmes engendrent une forte inertie à différents niveaux notamment dans :

- Le choix de primitives pertinentes décrivant la variabilité de la morphologie des caractères, sachant que certaines caractéristiques topologiques sont sensibles à la dégradation, notamment les points diacritiques et les boucles.
- La méthode de segmentation en caractères ou même en pseudo-mots (qui peuvent se chevaucher surtout dans le cas du manuscrit).

Tous ces problèmes et bien d'autres, se trouvent accentués dans le cas du manuscrit où d'autres facteurs interviennent (variabilité intra et inter-scripteurs, conditions de l'écriture, fusion de points diacritiques, chevauchement de pseudo-mots, graphismes inégalement proportionnés...).

Face à ces problèmes, la nécessité d'une modélisation robuste s'impose, les méthodes classiques de type statistique, structurelle, neuronale, etc étant peu efficaces pour prendre en considération toutes les variations morphologiques de l'arabe.

L'arabe s'écrit de droite à gauche avec un sens du tracé respecté (voir figure II.2) et la construction d'un mot se fait en collant les lettres une à une, de droite à gauche. Mais chaque lettre de l'alphabet s'écrit différemment selon sa position dans le mot. Les notions de lettre majuscule et lettre minuscule n'existent pas (l'écriture est donc monocamérale).

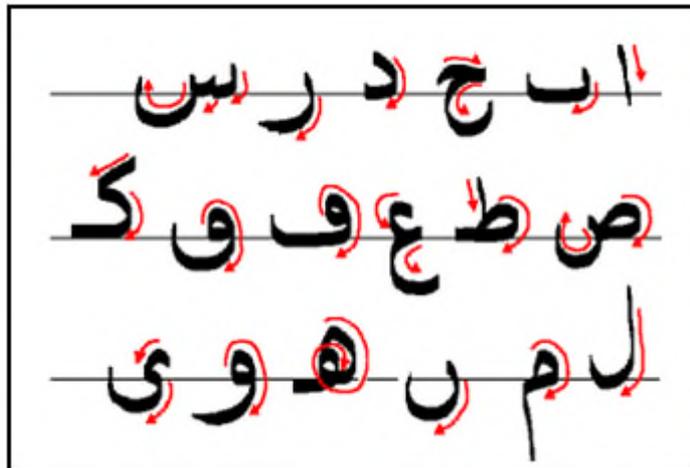


Figure II-2: Sens du d'écriture des lettres arabes

Le plus grand alphabet aujourd'hui contient maintenant environ 100 formes possibles. Donc, il existe une influence du scripteur sur les modèles d'écriture arabes et de divers modèles calligraphiques, certains sont montrés sur la figure II.3.



Figure II-3: Quelques styles de Calligraphie arabe

II.1.2.1 Signes diacritiques

L'écriture Arabe est riche en diacritique d'une façon générale, et plus particulièrement en points. Le terme 'signe diacritique' peut porter à confusion : dans certains travaux, seules les voyelles arabes sont appelées diacritiques. Dans d'autres travaux, en revanche, tous les signes secondaires sont appelés diacritiques, qu'il s'agisse des voyelles, des points ou des autres signes (chadda, madda, hamza, ...). C'est cette deuxième terminologie que nous employons ici : un signe diacritique est une composante secondaire d'une lettre, qui vient la compléter ou en modifier le sens.

II.1.2.1.1 Les points :

Nous comptons 15 lettres, parmi les 28 de l'alphabet qui comportent des points. Ces points apparaissent au dessus ou en dessous du caractère uniquement. Le nombre maximal de points que peut avoir une lettre est de trois points au-dessus du caractère, ou deux points en-dessous. Ces points permettent de différencier la prononciation des lettres Arabes comme montrée dans la figure II.4.

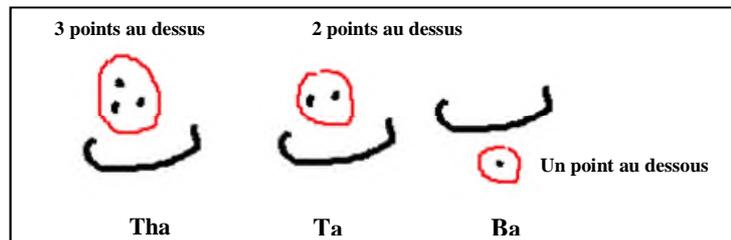


Figure II-4: Changement de prononciation de la même forme selon la position des points

Le tableau II.2 illustre la variabilité des styles d'écriture des points ou groupes de points en écriture manuscrite arabe. Un groupe de deux points peut ainsi s'écrire sous forme d'une seule, ou de deux composantes connexes.

Tableau II-2 : Variabilité des styles d'écriture des points

un point	.	'	-	
2 points	˘˘	˘˘	˘˘	˘˘
3 points	˘˘˘	˘˘˘	˘˘˘	˘˘˘

II.1.2.1.2 Les voyelles :

En arabe, les voyelles ne sont pas des lettres, mais des signes diacritiques associés aux lettres sur lesquelles ils s'appliquent (figure II.5).

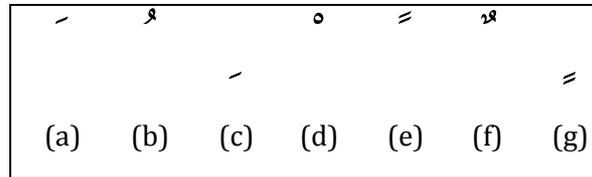


Figure II-5: Voyelles en arabe : (a) A, (b) OU, (c) I, (d) ~, (e) AN, (f) OUN, (g) IN.

En général on ne représente pas les voyelles signes, sauf dans les manuels scolaires.

L'absence de voyelles peut toutefois être source de confusions. Un mot peut avoir plusieurs voyellations possibles et par conséquent plusieurs catégories grammaticales.

Les voyelles peuvent parfois être mentionnées sur certaines lettres pour lever l'ambiguïté et faciliter la lecture. Mais en général, les scribes les omettent purement et simplement, et c'est au lecteur qu'est réservé le soin d'interpréter correctement le sens de la phrase en fonction du contexte.

II.1.2.1.3 Les autres signes diacritiques :

Les autres signes diacritiques sont la *hamza*, la *chaddet* la *madda*, (voir figure II.6). La chadda est une accentuation de la lettre (c'est l'équivalent d'une consonne doublée). Hamza et madda suivent des contraintes morpho syntaxiques plus complexes. [11]

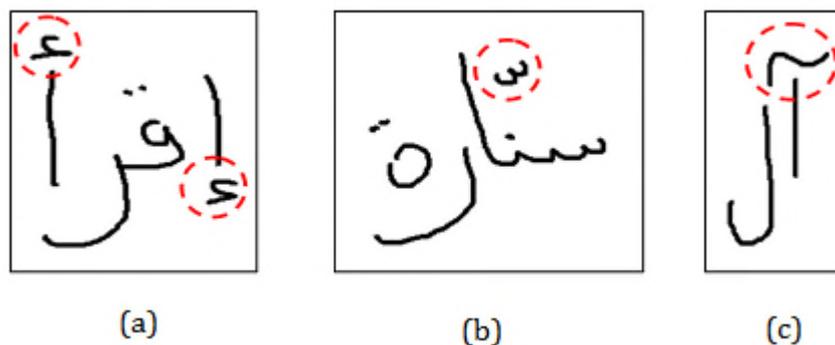


Figure II-6: Autres signes diacritiques : (a) hamza, (b) chadda, (c) madda

II.1.2.2 Une ou plusieurs composantes connexes par mot

L'écriture arabe est semi-cursive dans ses deux formes, imprimée et manuscrite. En effet, un mot arabe est une séquence d'entités connexes entièrement séparées

appelées *pseudo-mot*. Un mot peut être composé d'un ou de plusieurs pseudo-mots ; ceci est dû à la présence de caractères qui ne peuvent pas être attachés à leur successeur. Chaque pseudo-mot est une séquence de lettres liées, ce qui donne l'aspect de cursivité à cette écriture. Notons qu'un caractère isolé peut constituer un pseudo-mot à lui seul, comme le montre la figure II.7. Six lettres ne sont pas liées à leur successeur : و, ز, ر, ن, د, ا. Ces lettres introduisent donc une coupure dans le mot.

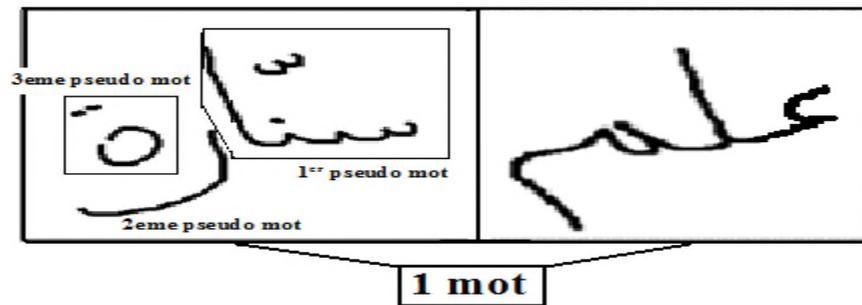


Figure II-7: (a) Le mot « drapeau » est composé d'un pseudo-mot, (b) Le mot « canapèche » est composé de 3 pseudo-mots.

II.1.2.3 Ligatures verticales

En écriture arabe, il n'y a pas de liaisons hautes comme le 'v' ou le 'o' en latin : les ligatures se situent au niveau de la ligne support de l'écriture (ligne de base). En revanche, les scribes sont libres de constituer certains groupes de deux ou trois lettres liées verticalement en début de pseudo-mot. Ce sont les ligatures verticales (figure II.8), leur complexité réside dans la segmentation en lettres.

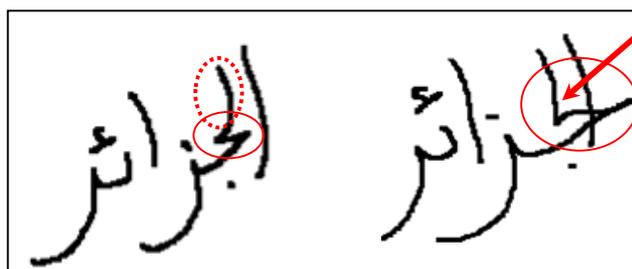


Figure II-8: Ligatures verticales et inversion de l'ordre du tracé

II.2 ETAT DE L'ART : BASES DE DONNEES & APPROCHES DE CLASSIFICATION

II.2.1 PRINCIPALES BASES DE DONNEES EXISTANTES

Dans [10] Atwell et al listent un ensemble d'outils pour l'analyse de corpus de textes

arabes appliqués au traitement automatique de la langue naturelle écrite ou orale. Des corpus de textes arabes conséquents sont disponibles pour l'analyse morphologique de la langue arabe.

En revanche, dans le domaine de la reconnaissance automatique de l'écriture manuscrite, les bases de données d'images annotées sont moins nombreuses.

Le domaine de la reconnaissance automatique de l'écriture arabe manuscrite a ainsi péché pendant de nombreuses années de l'absence d'une base de données de référence, qui permette des comparaisons objectives entre les différents systèmes. Des bases plus conséquentes mais payantes comme la base Kharma/Ahmed/Ward ou la base de chèques du CENPARMI sont apparues. Mais c'est la base IFN/ENIT, gratuite pour la recherche académique, qui s'est imposée comme la base de données de référence pour la comparaison des performances des systèmes de reconnaissance de l'écriture arabe manuscrite.

II.2.1.1 Kharma/Ahmed/Ward 1999

En 1999, N. Kharma et al [12] proposent une base de données, obtenue à partir des contributions de 500 étudiants :

- 37000 mots arabes
- 10000 chiffres, à la fois arabes (maghreb) et indiens (moyen orient)
- 2500 signatures
- 500 phrases

Les images sont disponibles à la fois en niveaux de gris et sous forme bancarisées. Cette base est disponible pour les chercheurs Canadiens, mais son accès est soumis à restrictions pour les chercheurs du reste du monde.

II.2.1.2 AHDB

En 2002, Almaadeed et al présentent une base de données collectées à l'aide d'une centaine de scripteurs [13]. Chacun des scripteurs était invité à écrire :

- Chacun des mots du vocabulaire des montants numériques.
- Trois montants numériques écrits en toutes lettres.
- Quelques lignes de texte libre.

Les 20 mots les plus utilisés par tous les scripteurs sont annotés manuellement.

II.2.1.3 CENPARMI

En 2003, Y. Al-Ohali et al [14] ont achevé la réalisation d'une base de données pour la reconnaissance de chèques arabes manuscrits.

Cette base se compose de 7000 images de chèques saoudiens issus de la pratique bancaire (voir figure II.9), scannées en niveaux de gris avec une résolution de 300 dpi [11]. Cet ensemble d'images a permis de mettre au point plusieurs bases de données:

- Une base de 1547 montants littéraux de chèques.
- Une base de 1547 montants numériques de chèques.
- Une base de 23325 pseudo-mots.
- Une base de 9865 chiffres indiens isolés.

Il serait également possible de mettre au point une base de données de dates à partir de ces images.



Figure II-9 : Exemple de chèque saoudien, et son montant littéral.

II.2.1.4 IFN/ENIT

M. Pechwitz et al introduisent la base IFN/ENIT en 2002[5]. Il s'agit d'une base de données d'images de noms de villes tunisiennes. Outre la séquence de lettres, sont également annotées la forme que prend chacune des lettres au sein du mot, la présence des signes diacritiques secondaires, et une approximation de la ligne de base.

411 scripteurs ont été mis à contribution, pour collecter environ 26400 noms de villes (dans un lexique de 937 villes) et plus de 210000 caractères.

Dans cette base, l'annotation est réalisée de telle sorte qu'une séquence de lettres contienne également l'information de la forme que prend chacune des lettres au sein du mot. (Voir figure II.10)

Cette base, par sa taille et sa disponibilité (gratuite pour des travaux académiques) s'est imposée comme la base de référence des travaux en reconnaissance de l'écriture manuscrite arabe. Elle sert de support à l'organisation des compétitions internationales dans le domaine de la reconnaissance de l'écriture arabe manuscrite

ICDAR ArabicHandwrittenCompetition, 2005 et 2007.

Image	حمام بياضة	رؤاد
Ground truth:		
Postcode	6132	2056
Global word	حمام بياضة	رؤاد
Character shape sequence	م_A ا_E م_M ل_L ح_B ة_E ض_B ا_E ي_M ب_B	و_A ل_L ر_A د_A ا_A
Baseline y1,y2	70,50	46,39
Baseline quality	B1 (B1=OK; B2=bad)	B1
Quantity of words	2	1
Quantity of PAWs	4	4
Quantity of characters	9	4
Writing quality	W1 (W1=OK; W2=bad)	W1

Figure II-10: Deux images et leurs annotations respectives dans la base IFN/ENIT. B pour Begin (début), M pour Middle (milieu), A pour Alone (isolé) et F pour Final (finale).

II.2.1.5 CEDARABIC

La base de données CEDARABIC [15] comprend 10 scripteurs différents. Chacun d'entre eux a écrit 10 pages complètes différentes de texte manuscrit. Chaque page comprend entre 150 et 200 mots, soit un total d'environ 20000 mots. Les documents sont scannés en 300 dpi. Une annotation complète des documents est faite manuellement : segmentation en mots, mais aussi séquence des lettres arabes, ainsi que la prononciation et la traduction en anglais.

Cette base de données est donc particulièrement adaptée pour des travaux sur la segmentation en mots et pour la recherche de mots clés dans une page d'écriture manuscrite en langue arabe.



Figure II-11: Exemple de segmentation en pseudo-mots qui porte à confusion [15].

II.2.1.6 Base de données latine UNIPEN

UNIPEN est un projet international pour la collection de données en ligne et des exemples pour l'écriture en-ligne, dans lesquelles un grand nombre d'universités et de compagnies. Le projet a eu une étape fermée, dans laquelle les environ 40 donateurs ont voulu réaliser des essais de test et d'apprentissage. Plusieurs compagnies et universités ont fournis des échantillons d'écriture en ligne d'UNIPEN. La première série de la base de données manuscrite en ligne d'UNIPEN est un "ensemble d'apprentissage", contenant assez de données pour extraire les configurations de test et d'apprentissage.

UNIPEN est une base de grande taille, très employée dans l'évaluation des performances des différents systèmes de reconnaissance d'écriture latine en ligne. [33]

La base UNIPEN est la base de référence pour l'élaboration et la comparaison de systèmes de reconnaissance d'écriture. Cette base contient des tracés de plus de 200 scripteurs. La difficulté de cette base est due principalement au nombre de scripteurs et donc aux nombreux allographes qu'ils emploient. La version R01/V06 est souvent usitée dans le domaine [34]. Cette base de données présente l'avantage de contenir un grand nombre de données dépaillées dans le tableau suivant :

Tableau II-3: *Taille de sous-corpus de la base UNIPEN*

Base	Nombre de Classes	Nombre total d'exemples	Nombre d'exemples en Apprentissage	Nombre d'exemples en Test
Chiffres	10	15635	10423	5212
Minuscules	26	52267	34844	17423
Majuscules	26	26605	17736	8869

Le tableau ci-dessous présente l'ensemble des répertoires fournis dans le CD-ROM de la base de données.[35]

Tableau II-4 : Description de la base de données UNIPEN

Répertoires	Description	Unité
1a	Chiffres isolés	char
1b	Lettres majuscules isolées	char
1c	Lettres minuscules isolées	char
1d	Symboles isolés (ponctuations etc.)	char
2	Caractères isolés, cas mixte	char
3	Caractères isolés dans le contexte des mots et du texte	char
4	Mots isolés imprimés, non mixés avec les chiffres et les symboles	word
5	Mots isolés imprimés, l'ensemble des caractères complet.	word
6	Le style mixte des mots (sans les chiffres et les symboles)	word
7	Mots isolés, aucun style, l'ensemble des caractères complet	word
8	Texte (minimum deux mots)	text

II.2.1.7 Base de données latine IRONOFF

La base de données IRONOFF (IReste ON/OFF database) est une base de données duales en-ligne et hors-ligne collectées et distribuées par LORIA. Elle contient un nombre important de caractères isolés, de chiffres et de mots en français et anglais au format UNIPEN. Cette base de données a été créée de telle sorte qu'un point en-ligne puisse être projeté sur sa position correspondante dans l'image scannée, et inversement chaque élément du tracé hors-ligne peut être temporellement indexé. [3]

Cette base a été acquise avec une tablette Wacom UltraPad, la résolution spatiale typique est de l'ordre de 300 points par pouce tandis que la fréquence d'échantillonnage des données est de l'ordre de 100 points à la seconde. Elle a été collectée auprès d'environ 700 scripteurs différents. Le tableau suivant montre un sous-corpus de la base de données contenant : les chiffres, lettres minuscules, lettres majuscules isolés.

Tableau II-5: Taille de sous-corpus de la base IRONOFF

Base	Nombre de Classes	Nombre total d'exemples	Nombre d'exemples en Apprentissage	Nombre d'exemples en Test
Chiffres	10	4108	3059	1510
Minuscules	26	11868	7952	3916
Majuscules	26	11879	7953	3926

II.2.1.8 Base de données arabe ADAB

La base de données ADAB (ArabicDATAbase) est développée pour améliorer la recherche et le développement des systèmes de reconnaissance du texte manuscrit arabe en ligne. Cette base de données est développée en coopération entre l'Institut des Technologies de communication (*Institute for Communications TechnologyIfN*) et l'Ecole Nationale d'Ingénieurs de Sfax (ENIS), le groupe de recherche (*Research Group on Intelligent Machines REGIM*), Sfax, Tunis.[29] La base de données dans sa version 1.0 contient 15158 mots arabes écrits par 130 scripteurs différents, la pluparts d'entre eux ont été sélectionnés de l'École Nationale d'Ingénieurs de Sfax (ENIS). Le texte écrit contient 937 noms de villes et villages tunisiens. Des outils spéciaux ont été développés pour l'acquisition et la vérification de données. Ces outils donnent la possibilité d'enregistrer les données écrites en ligne et d'autres informations concernant les scripteurs. Cette base de données est décomposée en 3 sous-corpus. Les détails à propos le nombre de fichiers, les mots, les caractères et les scripteurs pour chaque ensemble sont résumés dans la table suivante :

Tableau II-6: Sous-corpus de la base de données ADAB 1,2 et 3

<i>Sous-corpus</i>	<i>Fichiers</i>	<i>Mots</i>	<i>Caractères</i>	<i>Scripteurs</i>
1	5037	7670	40500	56
2	5090	7851	41515	37
3	5031	7730	40544	39
Somme	15158	23251	122559	132

II.2.2 DIFFERENTES SYSTEMES EXISTANTS:

II.2.2.1 Surveys existants

Les surveys suivants sont triées par ordre chronologique. Pour chacun d'entre eux, nous présenterons ses principaux apports, et les conclusions des auteurs.

II.2.2.1.1 ICDAR 1997 :

Dans [16]Adnan Amin analyse différentes approches pour la reconnaissance automatique de l'écriture arabe. Il présente les approches On-Line/Off-Line, ainsi que les approches Globale/Analytique. Il pose quelques unes des problématiques fondamentales de la reconnaissance de l'écriture arabe (alphabet, signes diacritiques, styles d'écriture, codage de l'écriture, pseudo-mots, ...). Il liste un nombre important de

systèmes, sans toutefois entrer dans le détail de leur fonctionnement.

Pour l'auteur, le sujet reste ouvert et de nombreuses voies d'amélioration sont possibles. L'évolution des performances des systèmes de reconnaissance de l'écriture arabe au cours des 10 dernières années montre qu'il avait raison.

II.2.1.1.2 CIFED 2000 :

Dans [17] Najoua Ben Amara et al s'intéressent aux approches markoviennes qui ont été mises en œuvre pour résoudre le problème de la reconnaissance de l'écriture arabe. Concernant l'écriture arabe manuscrite, Najoua Ben Amara et al passent notamment en revue les HMM-1D, les HMM-planaires.

Selon Ben Amara et al, les HMM-1D donnent des résultats encourageants, tout comme dans le cas de l'écriture latine. Les HMMs sont indépendants de l'alphabet considéré. Mais Ben Amara et al soulignent également les limitations classiques d'une modélisation linéaire, qui la encore, ne sont pas propres à l'écriture arabe.

Pour pallier ce problème, ils introduisent les PHMM, qui prennent en compte des variations bidirectionnelles de l'écriture. Les auteurs justifient en particulier ce choix sur l'arabe en raison du degré de difficulté supplémentaire que représente la présence de ligatures verticales. Ce dernier argument concernant la présence de ligatures verticales, n'est pas le plus pertinent. Les symboles qui correspondent aux ligatures verticales peuvent en effet être reconnus comme tels dans leur globalité, sans qu'il soit nécessaire de les segmenter.

II.2.2.1.3 PAMI 2006 :

Le Survey de Liana M. Lorigo et Venu Govindaraju [18], est le plus complet sur la reconnaissance de l'écriture arabe à l'heure actuelle.

L. M. Lorigo et V. Govindaraju passent en revue la plupart des problèmes spécifiques à la reconnaissance de l'écriture arabe. Ils consacrent un paragraphe à la reconnaissance de l'écriture imprimée, mais la plus grosse partie du document est réservée à la reconnaissance de l'écriture manuscrite. Ils découpent la chaîne de traitement d'un système de reconnaissance en fonctionnalités :

- Représentation
- Segmentation
- Extraction de primitives
- Moteur de reconnaissance, ou ils distinguent :
 - Système à base de règles

- Réseau de Neurones
- Modèles de Markov Cachées
- Systèmes hybrides.

En conclusion, les auteurs expliquent que des systèmes de reconnaissance de l'écriture arabe donnent des performances satisfaisantes sur des applications contraintes (petite taille de lexique, forme des mots relativement contrainte...).

L'avenir de la discipline se situe dans la capacité des systèmes à traiter de l'écriture libre, comme des courriers manuscrits. De telles applications nécessitent des modèles de langage développés, qui sont pour l'instant des voies largement inexplorées en langue arabe. L'intégration de contraintes morphologiques (analyse de la formation des mots à partir d'affixes, suffixes, racines) permettrait également de reconnaître des mots hors-vocabulaire.

II.2.2.1.4 SACH 2006:

Dans [19], A. Belaid et Ch. Choisy s'appuient sur le modèle de McClelland et Rumelhart[20] pour structurer leur analyse des systèmes de reconnaissance de l'écriture arabe manuscrite. (Voir figure II.12)

A. Belaid et Ch. Choisy insistent sur le fait que ce modèle est également applicable à la reconnaissance de l'écriture manuscrite arabe, à condition de lui rajouter un niveau intermédiaire : le niveau pseudo-mot.

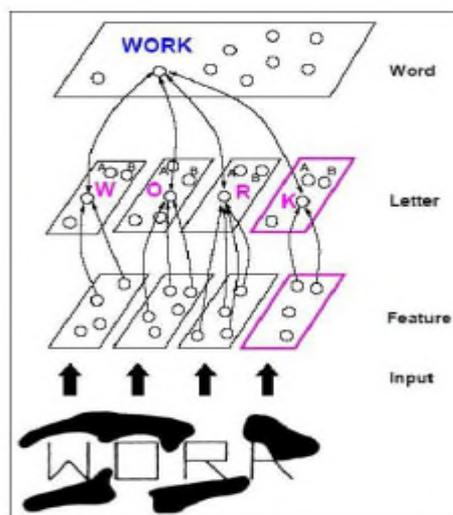


Figure II-12: Modèle de McClelland et Rumelhart

Les auteurs soulignent le fait que les primitives de bas niveau sont indépendantes du langage. En revanche, les primitives de plus haut niveau dépendent de la langue considérée, et nécessitent le développement de procédures spécifiques à la langue. Ces deux types de primitives sont complémentaires. Ils mettent en évidence la difficulté de la segmentation en mots, qui est selon eux l'une des raisons qui expliquent l'absence de système commercial de reconnaissance de l'écriture arabe manuscrite.

Ils mettent également en évidence le rôle crucial que jouent les pseudo-mots dans la langue arabe. Contrairement à l'écriture latine, en arabe l'entité élémentaire n'est pas le mot. Les approches globales devraient donc s'appuyer sur les pseudo-mots plutôt que sur les mots complets. Les approches analytiques devraient également exploiter cette particularité. Les auteurs déplorent le faible nombre de travaux qui exploitent cette particularité de la langue arabe dans leur chaîne de reconnaissance.

Les auteurs insistent également sur le fait qu'il est raisonnablement admis depuis longtemps dans la communauté de la reconnaissance de l'écriture cursive latine, que la segmentation en lettres est un problème mal posé. Rechercher une segmentation idéale en lettres dans un mot cursif est un problème qui n'a pas de solution. Or la segmentation d'un mot manuscrit arabe est un problème plus complexe que la segmentation d'un mot latin. Les auteurs déplorent donc la quantité de travaux qui s'attachent à optimiser la reconnaissance de lettres extraites à partir d'une segmentation manuelle, compte tenu qu'il semble illusoire qu'une segmentation automatique puisse fournir une segmentation en lettres idéale en entrée d'un reconnaiseur de caractères isolés.

Enfin, A. Belaid et Ch. Choisy [19] montrent un vif intérêt pour les systèmes hybrides qui, selon eux, semblent très prometteurs : ils combinent efficacement différents niveaux perceptifs, permettant ainsi de discriminer des mots sans avoir accès à une description complète. L'ajout d'informations locales à un système global permet d'étendre le vocabulaire en limitant les confusions. Les approches hybrides ne nécessitent pas une segmentation complète, et sont moins sujettes aux perturbations induites par les problèmes de perte d'information.

II.2.2.1.5 ISSPA 2007 :

Dans [21] Mohamed Cheriet fait une synthèse des différentes approches, avec la volonté d'ouvrir la réflexion sur de futures applications industrielles. M. Cheriet part

du constat suivant : les systèmes de reconnaissance de l'écriture manuscrite arabe, même s'ils obtiennent des performances encourageantes, n'ont pour l'instant fait leurs preuves que sur des données académiques, dans le cadre d'applications à l'environnement contraint. Il n'y a pas de système commercial de reconnaissance de l'écriture arabe manuscrite.

Il pose un certain nombre de questions ouvertes:

- Nécessité de procéder à une segmentation?
- Quel paradigme de reconnaissance utiliser?
- Quelle technique de reconnaissance utiliser?
- Faire un post-traitement, ou essayer d'intégrer les contraintes au sein de la chaîne de reconnaissance
- Redéfinir de nouvelles architectures, ou s'appuyer sur des techniques existantes?

M. Cheriet insiste également sur l'importance de l'analyse morphologique. Il s'interroge sur le meilleur endroit pour intégrer l'analyseur morphologique : au sein de la chaîne, ou à la fin en tant que post-traitement?

Il s'interroge également sur les moyens de constituer un lexique pour guider un système de reconnaissance à vocabulaire ouvert, et conclut par l'intérêt d'utiliser le Coran en tant que corpus, pour en dériver un lexique de manière automatique ou semi-automatique.

Il présente une étude de cas dans laquelle il combine trois niveaux perceptifs :

- Le niveau global, qui considère uniquement des indices visuels, qui sont issus des signes diacritiques et des tracés. Chaque mot est représenté par une série d'indices visuels qui sont traités par un système à base de Modèles de Markov Cachées.
- Un deuxième niveau, dans lequel les mots sont segmentés en graphèmes. Chaque graphème est alors transformé en un vecteur d'observation discret. Des Modèles de Markov Cachées sont entraînés pour décoder les séquences d'observations ainsi générées. A ce niveau, les signes diacritiques ne sont pas pris en compte, ce qui permet de réduire le nombre de modèles à estimer.

- Le niveau pseudo-mots, qui est un niveau intermédiaire entre les deux premiers. Des Modèles de Markov Cachées sont à nouveau utilisés, cette fois pour modéliser les probabilités de transition inter-pseudo-mots.

Une combinaison de ces trois modèles est ainsi effectuée pour évaluer la classe d'un mot inconnu. Pour terminer, M. Cheriet conclut sur l'intérêt de travaux de recherche qui viseraient à mieux intégrer le traitement de la langue naturelle dans une chaîne de reconnaissance de l'écriture arabe.

II.3 DIFFERENTES APPROCHES DE RECONNAISSANCE:

La reconnaissance est une tâche difficile à réaliser selon l'application demandée. C'est pour cela que les chercheurs ont développé plusieurs approches. On distingue l'approche statistique et l'approche structurelle. Une différence essentielle réside dans la représentation de la forme: vecteur de caractéristiques dans l'approche statistique, agencement de primitives pour l'approche structurelle. Nous allons rappeler quelques unes de ces approches couramment utilisées en les classant en 4 groupes: structurelle, statistique, neuronale et stochastique. Cette classification qui a été développée par [22] comprend une part d'arbitraire car celle-ci n'est pas unique.

II.3.1 APPROCHE STRUCTURELLE:

Les méthodes structurelles reposent sur la structure physique des caractères. Elles cherchent à trouver des éléments simples ou primitifs, et à décrire leurs relations. Les primitives sont de type topologiques telles que : une boucle, un arc... et une relation peut être la position relative d'une primitive par rapport à une autre. Dans cette approche on distingue plusieurs méthodes:

- **Méthode syntaxique:** ces méthodes sont directement issues de la théorie des langages formels. Elles se basent sur une grammaire formelle. Chaque caractère est représenté par une phrase dans un langage où le vocabulaire est constitué de primitives. La reconnaissance consiste à déterminer si la phrase de la description du caractère peut être générée par la grammaire. Cette méthode a été utilisée par Ramesh [23].
- **Comparaison des graphes:** cette méthode consiste à construire un graphe où les nœuds contiennent les primitives et les liens entre ces primitives. Ainsi la reconnaissance consiste à faire une mise en correspondance entre ces graphes et d'autres graphes représentant des caractères de référence construits lors de

la phase d'apprentissage. Cette méthode a été utilisée par Lebourgeois [24] et elle donne des résultats acceptables.

- **Comparaison de chaînes:** Les caractères sont représentés par des chaînes de primitives. La comparaison du caractère traité avec le modèle de référence, consiste à mesurer la ressemblance entre les deux chaînes et à se prononcer sur celui-ci. La mesure de ressemblance peut se faire par calcul de distance ou par examen de l'inclusion de toute ou une partie d'une chaîne dans l'autre.

II.3.2 METHODES DES K PLUS PROCHES VOISINS (KPPV):

L'algorithme KNN (K Nearest Neighbors) affecte une forme inconnue à la classe de son plus proche voisin en la comparant aux formes stockées dans une classe de références nommée prototypes. Il renvoie les K formes les plus proches de la forme à reconnaître suivant un critère de similarité. Une stratégie de décision permet d'affecter des valeurs de confiance à chacune des classes en compétition et d'attribuer la classe la plus vraisemblable (selon la métrique choisie) à la forme inconnue.

Cette méthode présente l'avantage d'être facile à mettre en œuvre et fournit de bons résultats. Son principal inconvénient est lié à la faible vitesse de classification due au nombre important de distances à calculer.

II.3.3 LES MACHINES A VECTEURS SUPPORTS (SVM):

Ces méthodes statistiques sont basées sur la définition des fonctions permettant de séparer partiellement ou totalement des classes représentées par les vecteurs paramètres de leurs échantillons.

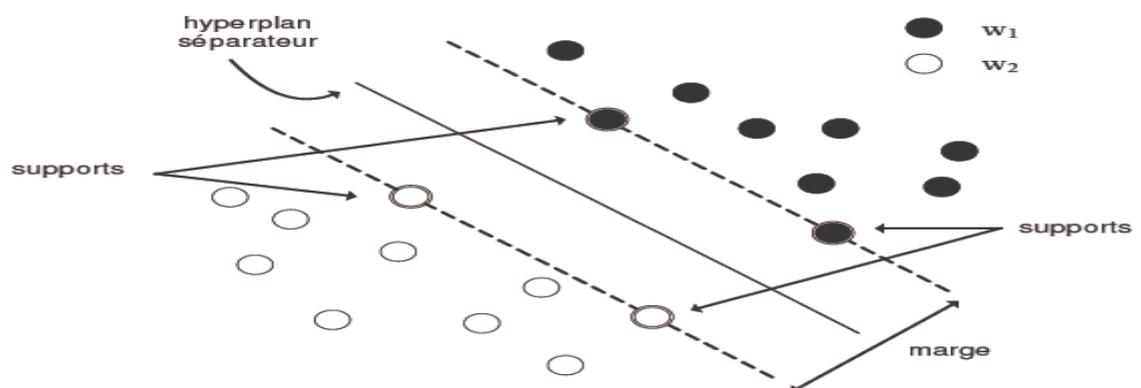


Figure II-13: Séparation souple par SVM: marge et hyperplan séparateur

Ces méthodes sont aussi appelées classifieurs à marge optimale ou encore séparateurs

à vaste marge. Il s'agit d'une famille d'algorithmes d'apprentissage mis en avant par le mathématicien Vladimir Vapnik [30].

L'idée principale est que deux classes (voir figure II.13, la classe des points noirs et la classe des points blancs) peuvent être linéairement séparées dans un espace de grande dimension $\Phi(x)$. Si les points d'apprentissage sont séparables, il existe une infinité d'hyperplans séparateurs.

On définit la marge comme étant la distance minimale entre la surface de décision et les exemples d'apprentissage (qui sont supposés être tous correctement classifiés par cette surface de décision). La difficulté est alors de trouver le meilleur hyperplan séparateur dans $\Phi(x)$ en utilisant les propriétés des fonctions à noyaux et de le transcrire dans un cas non séparable. Il s'agit d'un problème d'optimisation quadratique sous contraintes, ce qui est assez complexe (en termes algorithmiques) mais qui donne des garanties sur le temps de convergence (à un minimum global), contrairement à l'optimisation numérique d'une fonction de coût non-quadratique (comme pour des réseaux de neurones ou des mixtures de Gaussiennes). On veut maximiser la marge sous la contrainte que les exemples soient correctement appris.

Les SVM travaillent avec des données en dimension fixe, ils sont donc plus utilisés dans des applications restreintes à la reconnaissance de caractères ou de mots basée sur une segmentation explicite. L'inconvénient majeur de cette méthode est le coût important de mémorisation des vecteurs supports qui sont nombreux pour absorber la variabilité de l'écriture.

II.3.4 METHODES CONNEXIONNISTES:

L'objectif est d'améliorer les capacités de la classification en utilisant des modèles aux composants fortement connectés. Un modèle connexionniste est un réseau dont les nœuds sont interconnectés par des liens pondérés. Ainsi, dans un réseau connexionniste, l'information est traitée par un grand nombre de processeurs élémentaires, chacun étant relié à un nombre important d'autres processeurs. En général, le processeur est un neurone formel qui est un automate possédant n entrées réelles, son traitement consiste à effectuer à sa sortie le résultat d'une fonction de seuillage de la somme pondérée de ses entrées. Vu que la plus part des modèles reposent sur le neurone formel, on parle souvent des réseaux de neurones (RN). Ils sont considérés comme des discriminateurs linéaires complexes. La complexité de la

discrimination fonctionnelle que peut effectuer un réseau de neurones entraîne des conséquences:

- Adaptation a des surfaces de séparation complexes.
- Adoption d'une méthode d'apprentissage.

Afin de permettre au lecteur de mieux visualiser la structure d'un réseau de neurones, la Figure II.14 illustre, à l'aide d'un graphe de flot, un réseau de neurones :

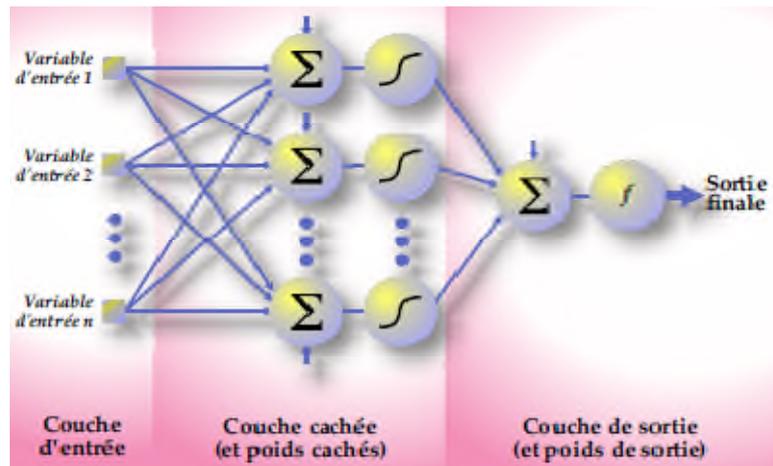


Figure II-14: Topologie d'un réseau de neurones.

Dans chaque unité de la couche cachée les variables sont combinées de façon linéaire. Le réseau de neurones applique une transformation non-linéaire à chacune de ces combinaisons. Finalement, les valeurs résultantes des unités cachées sont combinées de façon linéaire pour obtenir la valeur prédite.

Malgré la capacité des RN d'implanter des techniques discriminantes très efficaces, et leurs performances de classement, la durée d'apprentissage peut être très longue.

II.3.5 METHODES STOCHASTIQUES:

L'approche stochastique consiste à utiliser la modélisation par un processeur stochastique en prenant en compte la variabilité des caractères. Ainsi le caractère est considéré comme un signal continu observable dans le temps à différents endroits constituant des états d'observation. La reconnaissance consiste à comparer un échantillon de caractère de test à un modèle donné. Les paramètres des modèles sont calculés à l'aide des probabilités calculées de manière plus fine par apprentissage. Chaque modèle décrit ces états à l'aide des probabilités de transition d'état à état et de la loi de probabilité d'observation par état. Ainsi la comparaison consiste à chercher dans ce graphe d'états le chemin le plus probable correspondant à une suite

d'éléments observés dans la chaîne d'entrée. Ces méthodes sont robustes et fiables du fait de l'existence de bons algorithmes d'apprentissage.

II.3.5.1 Modèles de Markov cachés (HMM):

Un HMM est défini par les données suivantes:

- Un ensemble d'états
- Un ensemble de symboles observables dans chaque état
- Une matrice des probabilités de transitions
- Une matrice des probabilités d'observation
- Un ensemble de densités de probabilité initiale.

Ce modèle utilise essentiellement des données unidimensionnelles, ce qui a permis leur application directe en traitement de la parole, et ensuite pour l'OCR.

L'inconvénient est que les HMMs sont monodimensionnels alors que les images de caractères sont en dimension 2, ce qui donne des résultats parfois peu satisfaisants pour une application directe des HMMs.

II.3.5.2 Réseaux Bayésiens statiques et dynamiques:

Un réseau bayésien statique (BN) [25] est un mariage entre la théorie des graphes et la théorie des probabilités. Ainsi un réseau bayésien est constitué d'un graphe acyclique orienté dont les nœuds sont des variables aléatoires qui peuvent avoir un nombre discret d'états possibles ou dont les valeurs sont continues selon une loi continue, et d'un ensemble de distributions locales de probabilité qui sont les paramètres du réseau. Pour chaque nœud on dispose d'une table de probabilités conditionnelles. Récemment, les réseaux bayésiens sont appliqués dans le domaine de l'écrit. Nous pouvons citer les travaux de Cho dans lesquels il modélise les caractères on-line par des réseaux bayésiens: les variables qui représentent les positions des points caractéristiques, sont toutes observés. Souafi dans sa thèse [26] utilise un réseau bayésien classifieur pour la classification automatique de documents (apprentissage des structures des BNs en utilisant la méthode de propagation génétique). Enfin les réseaux bayésiens ont été utilisés pour la vérification automatique des signatures par Xiao et Leedham [27] (modélisation des dépendances entre composants de la signature).

Les réseaux bayésiens dynamiques (DBN) sont une extension des réseaux bayésiens statistiques et sont également appliqués à la reconnaissance automatique de la parole.

Daoudi dans [28] propose une approche multi-bandes ou le réseau bayésien permet une interaction entre sous-bandes du signal de parole.

II.3.6 LES CONTRIBUTIONS ORIGINALES:

- Une nouvelle représentation de caractères Arabes en-ligne basée sur les distributions empiriques de tangentes et de différences de tangentes à des points régulièrement espacés sur le signal du caractère. Cette représentation est motivée et adaptée à la nature topologique des caractères Arabes, riche en changement de direction.
- Un classificateur Bayésien basé sur l'estimation des densités de probabilité de classes par maximum d'entropie. Ceci fait intervenir le développement des aspects d'estimation des paramètres des densités de probabilité des classes, qui sont gibbsiennes, et des aspects d'estimation des fonctions de partition des différentes classes.
- Une combinaison simple de classificateurs bayésiens qui améliore les taux de reconnaissance.

Un classificateur hybride neuronal-probabiliste basé sur l'estimation par mémoires associatives de densités de probabilité gibbsiennes de classes d'entropie maximale.

II.4. Conclusion

L'écriture arabe est une écriture cursive qui présente l'avantage d'une liaison favorisant la fluidité et la rapidité du geste. Mais une des premières difficultés est la variabilité d'écriture entre scripteurs. La difficulté surmontée pour le traitement du caractère est sa forme et sa position dans le mot. La reconnaissance de l'écriture arabe manuscrite est en soi une tâche qui est fondamentalement plus complexe que la reconnaissance de l'écriture latine manuscrite cursive à cause des spécificités de l'écriture arabe qui présente un certain nombre de difficultés supplémentaires.

Plus nous avons présenté un état de l'art relatif à notre tâche de reconnaissance de manuscrit en-ligne. Après avoir introduit par la citation de quelques travaux du manuscrit en ligne à base de référence.

CHAPITRE III: LES RESEAUX DE NEURONES A CONVOLUTION (TDNN)

III.1 INTRODUCTION:

Le but de ce chapitre est de présenter la pierre angulaire de notre système de reconnaissance, en l'occurrence le TDNN et d'analyser l'impact des paramètres de ce réseau sur une application de reconnaissance de caractères isolés en matière de résultats.

Mais avant, nous introduirons dans un premier temps quelques propriétés sur les réseaux de neurones et les architectures présentes dans la littérature.

La reconnaissance du fait que le cerveau fonctionne de manière entièrement différente de celle d'un ordinateur conventionnel a joué un rôle très important dans le développement des réseaux de neurones artificiels. Les travaux effectués pour essayer de comprendre le comportement du cerveau humain ont menés à représenter celui-ci par un ensemble de composants structurels appelés neurones, massivement interconnectés entre eux. Le cerveau humain en contiendrait plusieurs centaines de milliards, et chacun de ceux-ci serait, en moyenne, connecté à dix mille autres. Le cerveau est capable d'organiser ces neurones, selon un assemblage complexe, non-linéaire et extrêmement parallèle, de manière à pouvoir accomplir des tâches très élaborées. Par exemple, n'importe qui est capable de reconnaître des visages, alors que c'est là une tâche quasiment impossible pour un ordinateur classique. C'est la tentative de donner à l'ordinateur les qualités de perception du cerveau humain qui a conduit à une modélisation électrique de celui-ci. C'est cette modélisation que tentent de réaliser les réseaux de neurones artificiels.

III.2 LES RESEAUX DE NEURONES ARTIFICIELS(RNA)

III.2.1 LE NEURONE FORMEL:

Chaque neurone réalise une opération très simple, qui est en fait une somme pondérée de ses entrées. Le résultat est comparé à un seuil et le neurone devient excité si ce seuil est dépassé. L'information contenue dans le cerveau est représentée par les poids donnés aux entrées de chaque neurone. Du fait du grand nombre de neurones et de leurs interconnexions, ce système possède une propriété de tolérance aux fautes. Ainsi, la défectuosité d'un élément mémoire (neurone) n'entraînera aucune perte réelle d'information, mais seulement une faible dégradation en qualité de toute l'information

contenue dans le système. C'est pourquoi nous pouvons reconnaître le visage d'une personne, même si celle-ci a vieilli, par exemple.

La première étude systématique du neurone artificiel est due au neuropsychiatre McCulloch et au logicien Pitts[31] qui, s'inspirant de leurs travaux sur les neurones biologiques, proposèrent en 1943 le modèle suivant :

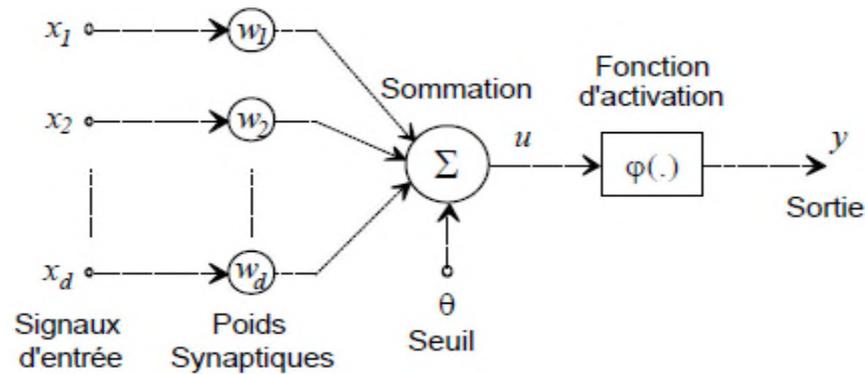


Figure III-1: Le modèle du neurone formel

Ce neurone formel est un processeur élémentaire qui réalise une somme pondérée des signaux qui lui parviennent. La valeur de cette sommation est comparée à un seuil et la sortie du neurone est une fonction **non linéaire** du résultat:

$$u = \sum_{j=1}^d w_j x_j - \theta \quad (\text{III-1})$$

$$y = \varphi(u) \quad (\text{III-2})$$

En prenant la convention de noter par $\underline{X} = [-1x_1 \dots \dots x_d]^T$ le vecteur d'entrées augmenté, et par $\underline{W} = [\theta w_1 \dots \dots w_d]^T$ le vecteur de poids augmenté, il vient:

$$y = \varphi(\underline{W}^T \underline{X}) \quad (\text{III-3})$$

Dans le modèle original de McCulloch et Pitts, la non-linéarité était assurée par la fonction seuil de Heaviside, définie par:

$$H(u) \triangleq \begin{cases} 1 & \text{si } u > 0 \\ 0 & \text{sinon} \end{cases} \quad (\text{III-4})$$

En place de la fonction de Heaviside, il est également possible de choisir la fonction de signe, définie comme suit:

$$\text{sgn}(u) \triangleq \begin{cases} 1 & \text{si } u > 0 \\ -1 & \text{sinon} \end{cases} \quad (\text{III-5})$$

III.2.1.1 Définition d'un RNA:

C'est un ensemble de neurones formels associés en couches et fonctionnant en parallèle. Il a la capacité de stocker de la connaissance empirique (dans les poids synaptiques, à l'aide des processus d'apprentissage) et de la rendre disponible à l'usage. On distingue généralement trois paradigmes d'apprentissage: supervisé, non supervisé et hybride.

La propagation est alors le phénomène qui se produit lorsque l'on entre des données dans le réseau, et que l'information se propage de neurone en neurone jusqu'aux points de sortie. Inversement, l'apprentissage s'effectue par retro-propagation. Dans le cas d'un apprentissage supervisé, on fait une propagation avec l'entrée désirée, puis on impose à la sortie. On peut alors calculer approximativement l'erreur commise par les derniers neurones. Après correction des poids, on transmet l'erreur correspondante aux neurones précédents. Ainsi de suite, on remonte jusqu'à la couche d'entrée en modifiant les poids des neurones. C'est la retro-propagation.

L'apprentissage non-supervisé consiste à ne pas imposer de sortie lors de l'apprentissage. Si deux entrées sont de la même classe, elles devraient par la suite produire la même sortie.

III.2.2 L'ARCHITECTURE DE BASE : LE PERCEPTRON MULTI COUCHES (PMC)

III.2.2.1 Structure d'un PMC:

Un perceptron multi couches comporte généralement trois types de couches: la couche d'entrée, la ou les couches cachées et la couche de sortie. Chaque neurone d'une couche est connecté à tous les neurones de la couche précédente, on dit que le réseau est complètement connecté.

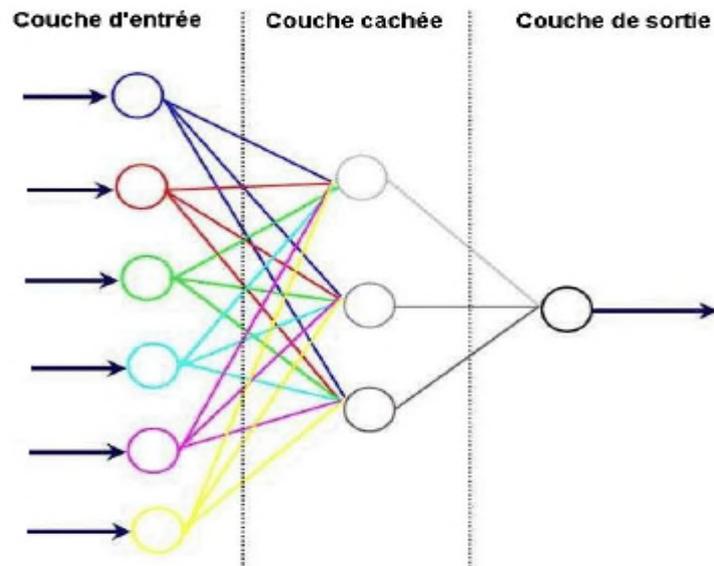


Figure III-2: architecture de base d'un PMC

Le schéma suivant représente un PMC à trois couches. La couche d'entrée comporte 6 caractéristiques. La couche cachée contient trois neurones tandis que la couche de sortie ne contient qu'un seul neurone. La couche cachée a pour but d'extraire de l'espace d'entrée de l'information pertinente pour résoudre le problème de décision

III.2.2.2 Apprentissage:

Pour pouvoir attribuer à un élément une classe, il faut tout d'abord apprendre à discriminer cette classe et être capable de généraliser la problématique avec un nombre d'échantillons réduit, car même avec un contexte multi-scripteur (comme le notre, nous y reviendrons ci-dessus dans le chapitre suivant), il est quasiment impossible de récupérer tous les prototypes à cause de la grande variabilité du signal d'écriture et cela pour un même scripteur.

Nous utiliserons un apprentissage supervisé, notre base de données intitulée NOUN-DATABASE (chapitre suivant) est partagée en deux parties. La première sous-base, appelée base d'apprentissage permet comme son nom l'indique d'apprendre les poids du réseau de neurones. La deuxième sous-base, appelée base de test sert à ajuster les paramètres de l'algorithme d'apprentissage en matière de nombre de couches cachées, nombre de neurones, pas de gradient, ...). La méthode $\frac{2}{3} \frac{1}{3}$ est fréquemment utilisée, soit plus de 60% des échantillons est utilisée pour la base d'apprentissage et le reste pour la base de teste.

Suivant la complexité du problème d'optimisation, plusieurs types de méthodes peuvent être envisagés, il y'a celles du premier ordre :

- Rétro propagation du gradient
- Rprop.

Et celles du deuxième ordre

- Les gradients conjugués.
- Levenbergmarquard.

III.2.2.2.1 La Rétro Propagation du Gradient

C'est un algorithme d'apprentissage pour un réseau de type PMC Pour un ensemble d'éléments (d'apprentissage), trouver les poids du réseau PMC qui fournissent la réponse attendue pour chacun des éléments.

Le traitement d'un élément se fait en 2 passes :

Passé avant : on présente l'élément à l'entrée du réseau de neurones et on obtient le résultat en sortie

Passé arrière : on calcule l'erreur de ce résultat par rapport au résultat attendu pour chaque neurone de sortie et on répercute cette erreur aux neurones des couches internes avec un algorithme de type « gradient-descendant-GD ».

Les poids ne sont pas adaptés après chaque élément, on se contente d'additionner les gradients pour chaque neurone, afin d'avoir une vue globale de l'erreur pour l'ensemble. A la fin d'une époque, les poids sont adaptés.

Le gradient pour un neurone est l'erreur relative à ce neurone. Il peut en quelque sorte être vu comme la contribution du neurone à l'erreur globale

A chaque passé arrière, on calcule le gradient de chaque neurone, en commençant par ceux de la couche de sortie (car les gradients des couches inférieures se calculent à partir des gradients des couches supérieures).

La rétro-propagation du gradient utilise l'erreur quadratique, a_i est l'activation et d_i est la valeur désirée du neurone de sortie i , qui est donnée par :

$$E = \frac{1}{2} \sum_i (d_i - a_i)^2 \quad (\text{III-6})$$

Les poids sont modifiés en y ajoutant la valeur ΔW_{ij} où h est le pas d'apprentissage

$$\Delta W_{ij} = -h \frac{\partial E}{\partial W_{ij}} \quad (\text{III-7})$$

Le calcul du gradient d'erreur par rapport aux poids :

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial net_i} \cdot \frac{\partial net_i}{\partial w_{ij}} \quad (\text{III-8})$$

Où net_i est la somme de l'activité pondérée qui converge sur le neurone i .

Pour la couche externe (la couche de sortie), cette valeur se calcule de la manière suivante :

$$\frac{\partial E}{\partial net_i} = \frac{\partial}{\partial net_i} \left[\frac{1}{2} \sum_i (d_i - f(net_i))^2 \right] = -(d_i - a_i) \cdot f'(net_i) \quad (\text{III-9})$$

Où $f(net_i)$ est la fonction d'activation du neurone i .

$$\frac{\partial net_i}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}} \left[\sum_k w_{ik} \cdot a_k \right] = a_j \quad (\text{III-10})$$

Pour les neurones de la couche interne (la couche cachée) :

$$\frac{\partial E}{\partial net_i} = \left(\sum_k \frac{\partial E}{\partial net_k} \cdot \frac{\partial net_k}{\partial a_i} \right) \cdot \frac{\partial a_i}{\partial net_i} = \left(\sum_k \frac{\partial E}{\partial net_k} \cdot w_{kj} \right) \cdot f'(net_i) \quad (\text{III-11})$$

L'indice k parcourt tous les neurones des couches supérieures à celle du neurone i . A partir de ce calcul, on détermine la modification à apporter aux poids du réseau :

$$\Delta w_{ij} = -h \frac{\partial E}{\partial net_i} \cdot \frac{\partial net_i}{\partial w_{ij}} \quad (\text{III-12})$$

Et les liens du réseau doivent être corrigés comme suit :

$$\Delta w_{ij} = -h \cdot e_i \cdot a_i \quad (\text{III-13})$$

Où e_i représente la valeur accréditée du neurone i de l'erreur de j .

Pour la couche externe, l'erreur se calcule :

$$e_i = -(d_i - a_i) \cdot f'(net_i) \quad (\text{III-14})$$

Pour l'autre couche, elle se calcule :

$$e_i = \left(\sum_k e_k \cdot w_{ki} \right) \cdot f'(net_i) \quad (\text{III-15})$$

Sachant que e_k est l'erreur précédemment calculée pour le neurone k .

III.2.2.2 Rétropropagation du gradient "résistante" : Rprop

La rétropropagation du gradient "résistante" est un algorithme d'apprentissage pour les perceptrons multicouches [36]. Cet algorithme est un des plus rapides et précis, de plus, il a pour qualité d'être relativement robuste. Cet algorithme reste un algorithme d'apprentissage du premier ordre et donc sa complexité augmente linéairement en fonction de la topologie du réseau.

Contrairement à l'algorithme classique de rétropropagation du gradient, l'algorithme Rprop n'utilise pas directement la dérivée de l'erreur ($\delta E^{(l)}$) pour mettre à jour les poids du réseau (w_{ij}). Dans l'algorithme Rprop, seul le signe de la dérivée de l'erreur est utilisé pour déterminer si les poids sont diminués ou augmentés. De plus, la modification des poids est réalisée en utilisant une valeur indépendante de l'erreur $\Delta_{ij}^{(l)}$ qui évolue au cours du processus d'apprentissage à l'aide d'un facteur constant (strictement inférieur η^- à 1 pour la diminution et strictement supérieur η^+ dans le cas contraire). Cette valeur de modification est bornée ($\Delta_{\min} ; \Delta_{\max}$).

Christian Igel et Michael Husken ont publié une évolution de cet algorithme [37]. Cette évolution permet d'annuler une modification de poids si l'effet de celle-ci est négatif. L'algorithme Rprop ainsi modifié est nommé iRprop+ et est détaillé dans l'algorithme III-3.

Comme dans le cas de l'algorithme classique Rprop, seul le signe de la dérivée de l'erreur est utilisé pour effectuer la mise à jour des poids. Dans le cas où la dérivée est strictement négative, c'est à dire que l'erreur se réduit, la valeur de modification des poids est augmentée et les poids du réseau sont modifiés afin de minimiser l'erreur restante. Dans le cas où l'erreur est augmentée, la constante de modification des poids est diminuée et les différents poids du réseau sont remis à leur valeur précédente. Enfin, si l'amélioration est constante, la constante de modification reste inchangée et les poids sont mis à jour afin de minimiser l'erreur restante.

<p>Pour tout w_{ij} Faire</p> <p>Si $\frac{\delta E^{(t-1)}}{\delta w_{ij}} \cdot \frac{\delta E^{(t)}}{\delta w_{ij}} > 0$ Alors</p> <p>$\Delta_{ij}^{(t)} \leftarrow \min(\Delta_{ij}^{(t-1)} \cdot \eta^+, \Delta_{max})$</p> <p>$\Delta w_{ij}^{(t)} \leftarrow -\text{sign}\left(\frac{\delta E^{(t)}}{\delta w_{ij}}\right) \cdot \Delta_{ij}^{(t)}$</p> <p>$w_{ij}^{(t+1)} \leftarrow w_{ij}^{(t)} + \Delta w_{ij}^{(t)}$</p> <p>Sinon Si $\frac{\delta E^{(t-1)}}{\delta w_{ij}} \cdot \frac{\delta E^{(t)}}{\delta w_{ij}} < 0$ Alors</p> <p>$\Delta_{ij}^{(t)} \leftarrow \max(\Delta_{ij}^{(t-1)} \cdot \eta^-, \Delta_{min})$</p> <p>Si $E^{(t)} > E^{(t-1)}$ Alors</p> <p>$w_{ij}^{t+1} \leftarrow w_{ij}^t - \Delta w_{ij}^{(t-1)}$</p> <p>Finsi</p> <p>$\frac{\delta E^{(t)}}{\delta w_{ij}} \leftarrow 0$</p> <p>Sinon Si $\frac{\delta E^{(t-1)}}{\delta w_{ij}} \cdot \frac{\delta E^{(t)}}{\delta w_{ij}} = 0$ Alors</p> <p>$\Delta w_{ij}^{(t)} \leftarrow -\text{sign}\left(\frac{\delta E^{(t)}}{\delta w_{ij}}\right) \cdot \Delta_{ij}^{(t)}$</p> <p>$w_{ij}^{(t+1)} \leftarrow w_{ij}^{(t)} + \Delta w_{ij}^{(t)}$</p> <p>Finsi</p> <p>Fin pour</p>
--

Figure III-3 : Algorithme de iRprop+

Néanmoins, il existe d'autres méthodes d'apprentissage du second ordre, elles sont dites du second ordre parce qu'elles prennent en considération la dérivée seconde de la fonction de coût. Nous présentons ci dessous celles que nous avons mises en œuvre dans notre travail (chapitre IV), et dont nous comparons les performances lors de l'étude de nos modèles.

III.2.2.2.3 Méthode de Levenberg-Marquart (2^{ème} ordre):

Parmi les algorithmes de la famille quasi-Newton, la méthode de LEVENBERGMARQUARDT est un standard pour l'optimisation de l'erreur quadratique due à ses propriétés de convergence rapide et de robustesse. Elle s'appuie sur les techniques des moindres carrés non-linéaires et de l'algorithme de GAUSS-NEWTON à voisinage restreint.

L'algorithme de Levenberg-Marquart(LM) est un algorithme itératif, qui consiste à modifier les paramètres proportionnellement au gradient de la fonction coût total. Il a été appliqué pour l'apprentissage des réseaux de neurones dans plusieurs travaux de recherche. Cet algorithme adopte une direction de déplacement qui n'est plus la direction du gradient, mais une transformation linéaire appropriée du gradient du coût total. Ca consiste à modifier les paramètres du réseau selon la formule suivante:

$$W(i) = W(i-1) - [H(W(i-1)) + \lambda_i I]^{-1} \nabla J(W(i-1))$$

Où λ_i est un paramètre constant strictement supérieur à zéro et H est la matrice hessienne ou la dérivée seconde par rapport aux paramètres.

Cet algorithme s'applique à la fonction de coût quadratique parce que son hessien peut être approché par la relation suivante:

$$H = Ja^T Ja$$

Où Ja est la matrice jacobienne (ou le jacobien).

L'algorithme de Levenberg-Marquardt est détaillé par l'algorithme ci-dessous:

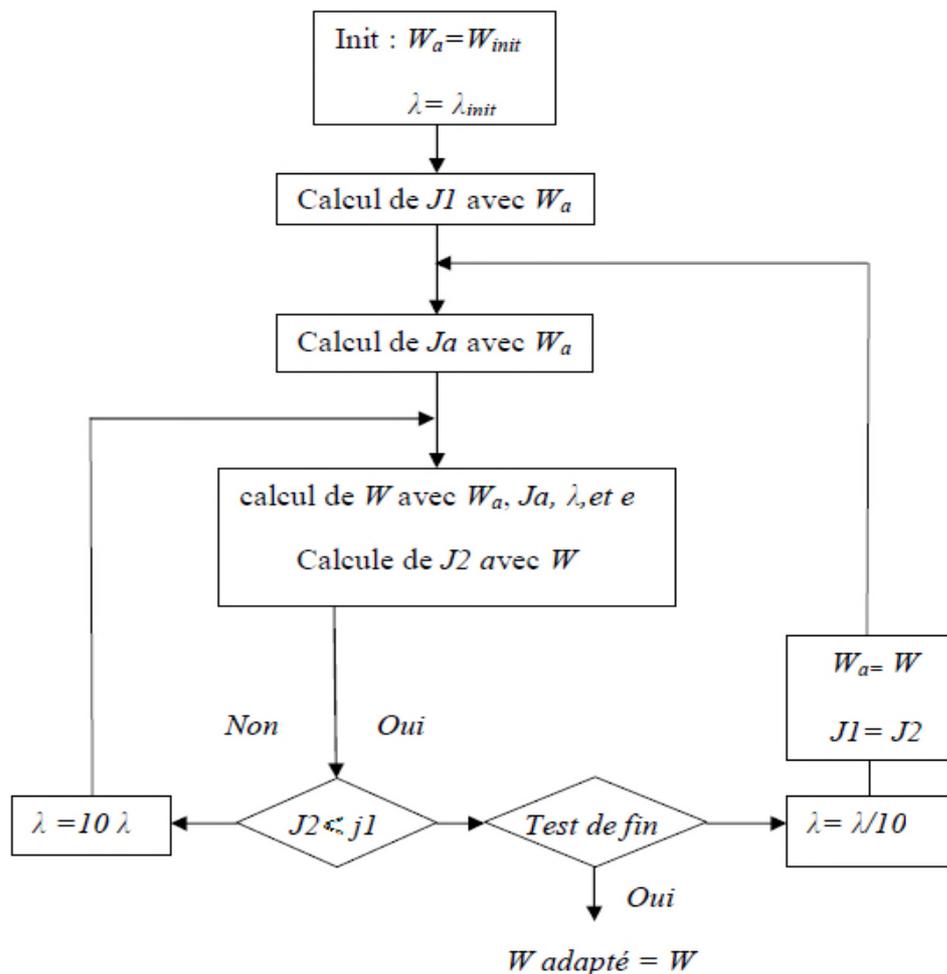


Figure III-4: L'algorithme de Lavenberg-Marquardt.

III.2.2.2.4 Méthode d'apprentissage par algorithme du gradient conjugué

Cette méthode initiée par Hestenes and Stiefel en 1952, est une méthode itérative qui se base sur la recherche de directions successives permettant d'atteindre la solution exacte d'un système de matrice symétrique et définie positif, elle est intéressante pour les RNA à un grand nombre de neurones : convergence rapide et est modeste en mémoire.

III.2.2.3 Le paramétrage

III.2.2.3.1 Initialisation des poids et choix du pas

L'initialisation des poids peut avoir une influence grandiose dans l'apprentissage du réseau. Les poids doivent être choisis d'une façon aléatoire mais sans oublier que ce choix doit être compris dans la partie linéaire de la sigmoïde pour que les gradients seront suffisamment importants pour permettre un apprentissage correct.

III.2.2.3.2 Architecture d'un réseau de neurones

La capacité d'un RN dépend du nombre des poids du réseau. Pour être capable d'apprendre de décision complexe, le PMC doit avoir un nombre de couches suffisant, un minimum de deux couches cachées ainsi qu'un nombre suffisant de neurones sur les couches. Cependant, la qualité des résultats obtenus ne dépend pas seulement de cela, d'autres paramètres rentrent en jeu tel que: l'apprentissage et les paramètres d'entrée. Il n'existe pas une structure unique du réseau, différentes topologies en termes de taille et type de connexions peuvent conduire à des performances plus ou moins similaires. L'optimisation d'un RN conduit jouer sur ses paramètres (initialisation des poids, choix de la fonction d'activation, nombre de neurones et nombre de couches cachées) et cela dans un but unique, celui d'atteindre des performances maximale.

La définition de l'architecture d'un RN repose sur deux méthodes principales, la première est la méthode empirique (celle choisie dans notre travail), qui consiste à comparer expérimentalement les performances obtenues pour différentes architectures, et de retenir celle qui assure un meilleur compromis performance\ coût architectural. Aucun théorème ne définit le nombre de couches cachées ni celui du nombre de neurones dans chaque couche cachée, néanmoins ce nombre varie selon la précision souhaitée. Le nombre de neurones de la couche de sortie est égale au nombre de classes qu'on veut distinguer. La méthode empirique permet aussi d'intégrer des connaissances préalables des données comme les connaissances temporelles ou spatiales pour les réseaux à convolution que nous allons présenter dans ce chapitre.

La deuxième méthode repose sur les algorithmes d'optimisation automatique, on peut citer les algorithmes de croissances qui effectuent l'apprentissage avec au départ une architecture très petite et de l'accroître par la suite. Les algorithmes de dégénérescence de réseau de neurones partent d'un réseau surdimensionné que l'on simplifie au cours de l'apprentissage en réduisant le nombre de neurones et de connexions.

III.3 LES RESEAUX DE NEURONES A CONVOLUTION:

Le RN représente un ensemble de connaissances en permettant un apprentissage automatique des poids avec un fort pouvoir de généralisation. Précédemment, nous avons mis en relief les principes d'un PMC qui s'appuient sur des entrées de dimension fixe et une vision globale du système. Les aspects temporels et 2D ne sont pas pris en compte, par contre dans une architecture à convolution, il est possible d'intégrer ces notions.

La vision locale du système dans un réseau à convolution est due à l'architecture dans son approche connexionniste, un neurone d'une couche est connecté à un ensemble de neurones de la couche précédente. Chaque neurone peut être vu comme une unité de détection d'une caractéristique locale.

Les réseaux de neurones utilisent les notions de zones réceptives locales et les poids partagés. Ces derniers permettent la réduction du nombre de paramètres dans le système facilitant ainsi la généralisation. Le concept de poids partagés se base sur le principe que dans le cerveau humain, des neurones détectent certains traits dans de petites régions de la rétine, essentiellement de la même manière dans toutes ces régions. Ces neurones calculent la même fonction sur des entrées différentes. De ce fait, il est nécessaire que ces neurones partagent les mêmes poids.

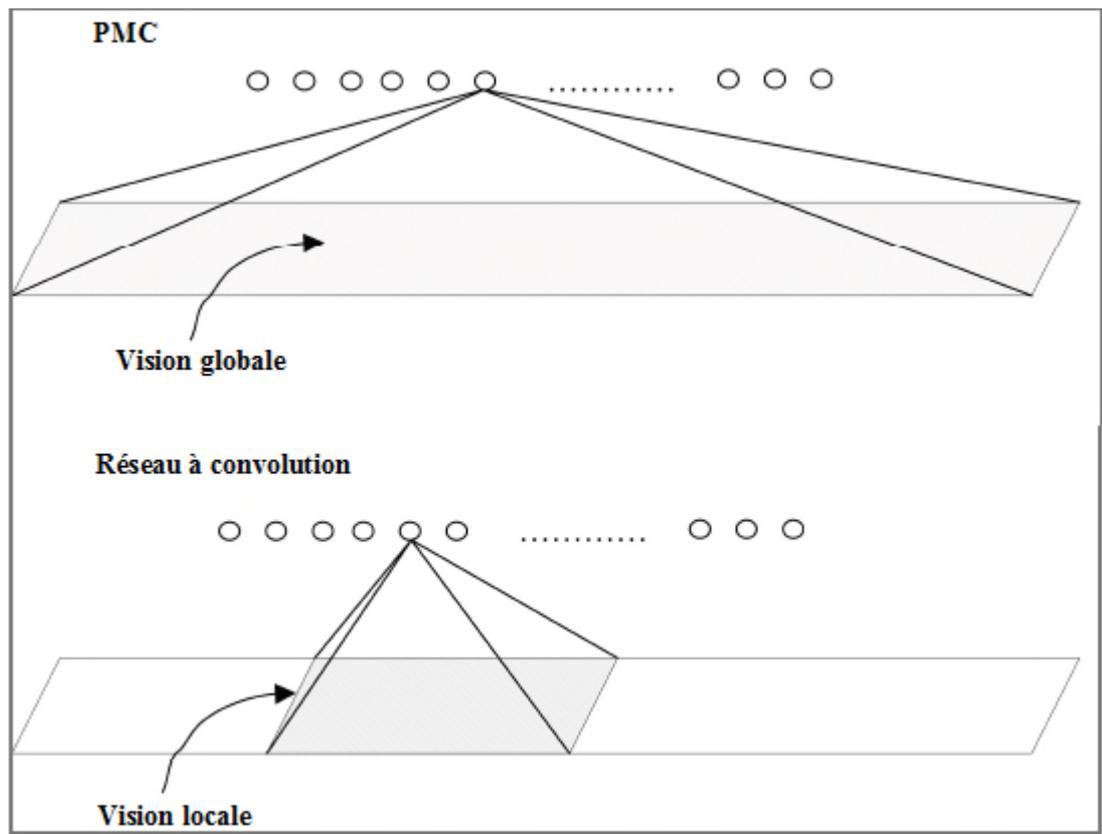


Figure III-5: Illustration des connexions dans un PMC et dans un RNC.

La reconnaissance des formes, qui a constitué un champ d'application privilégié des réseaux de neurones, concerne essentiellement des données statistiques. Un grand nombre d'applications possèdent cependant une dimension temporelle ne serait-ce que parce que les données d'entrées varient dans le temps. C'est également le cas d'un grand nombre d'activités humaines comme par exemple le langage, la vision ou le mouvement. Comment dès lors envisager dans ce contexte l'utilisation des réseaux de neurones pour l'apprentissage de telles activités sans qu'eux mêmes ne possèdent une dimension temporelle?

La représentation spatiale du temps est tout d'abord exposée. Un exemple caractéristique (TDNN) d'une telle approche est décrit ainsi que les avantages inhérents à ce type de modèle sont présentés

III.3.1 RESEAUX DE NEURONES ADAPTES AUX DONNEES SPATIOTEMPORELLES

De nombreuses applications, traitées par des réseaux de neurones, requièrent l'utilisation de données présentant des caractéristiques temporelles : filtrage, mémoire associative, séparation aveugle de sources, commande de processus, détection, etc. On parle de

données spatio-temporelles lorsqu' à un instant donné, plusieurs caractéristiques sont représentées. A titre d'exemple, un spectrogramme est considéré comme une donnée spatio-temporelle.

Les réseaux de neurones peuvent exploiter l'information temporelle de différentes manières. On peut citer :

- Une représentation spatiale de l'information temporelle,
- L'introduction de retards dans les neurones ou les connexions,
- L'utilisation de connexions récurrentes.

III.3.1.1 Représentation spatiale de l'information temporelle

La façon la plus immédiate de représenter le temps dans les réseaux de neurones est d'utiliser ce qui est appelée une représentation spatiale du temps. L'information temporelle que contiennent les données est alors transformée en une information spatiale, c'est-à-dire une forme qu'il s'agit de reconnaître. Dès lors les techniques de classification par réseaux de neurones habituellement employées deviennent applicables.

En utilisant des registres à décalage ou des lignes à retard, il est possible d'effectuer une transformation *temps-espace* de la donnée à caractère temporel. Cette transformation a pour but de représenter l'évolution de caractéristiques temporelle en entrée du réseau.

Ainsi, l'entrée du réseau étant un signal temporel $x(t)$ (figure III-6), ce signal passe à travers diverses unités de retard dont le nombre définit la largeur de la fenêtre d'observation.

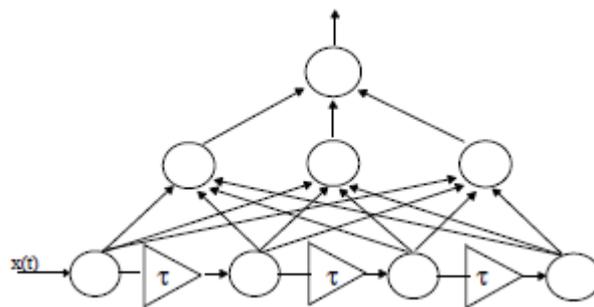


Figure III-6: Représentation spatiale de l'information en entrée d'un réseau de neurones

III.3.1.2 Les limites de la représentation spatiale

Présentation spatiale du temps, qui est caractérisée par l'emploi d'une métaphore spatiale du temps, présente plusieurs désavantages. Tout d'abord elle suppose l'existence d'une interface avec le monde extérieur dont le rôle est de retarder ou de retenir les données jusqu'au moment de leur utilisation dans le réseau. Sans considérer la probabilité de l'existence de tels mécanismes au niveau du neurone biologique, se pose de toute façon une question de logique: comment connaître l'instant où les données doivent être traitées?

Ceci nous amène au second désavantage qui représente le fait d'utiliser une fenêtre temporelle (ou des retards) de longueur finie et déterminée a priori soit par la plus longue information à traiter, soit en supposant la même longueur pour toutes les données. Dans certaines applications comme le traitement de la parole il serait préférable de disposer d'un réseau capable de traiter des informations de longueur variable. Enfin, c'est dans la nature même de la spatiale que se pose la difficulté de différencier une position temporelle d'une position temporelle absolue

III.3.1.3 Classification d'architectures de réseaux de neurones \ temps:

Dans la littérature, on trouve plusieurs architectures de réseaux de neurones statiques et dynamiques. Les réseaux de neurones statiques tels que les PMC et le SOM (Self Organizing Map) n'implémentent pas la notion temps. En revanche, on distingue deux grandes familles de réseaux de neurones avec la prise en compte d'une certaine représentation du temps : les réseaux à représentation externe du temps, et ceux à représentation interne. Les réseaux à représentation externe du temps admettent des informations sur les délais sur la couche d'entrée. En revanche, les réseaux de neurones à représentation implicite du temps utilisent des connexions bouclées afin de représenter cette notion de temps. La figure III.7 représente une classification de plusieurs réseaux de neurones statistiques et temporels.

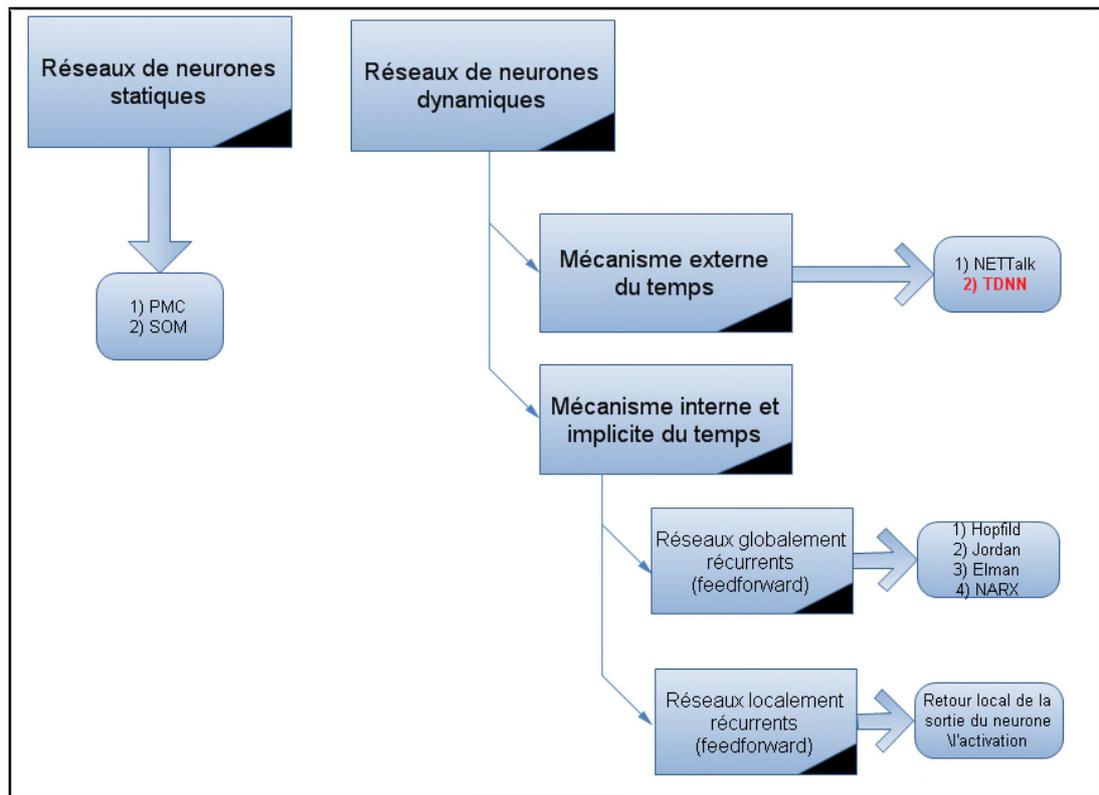


Figure III-7: Classification d'architectures de réseaux de neurones

Dans la partie des réseaux de neurones à mécanisme externes du temps, on dispose de deux exemples d'architectures qui sont NETTalk et le TDNN. L'aspect temporel du TDNN est implémenté en utilisant des retards temporels pour chaque neurone du réseau.

La deuxième partie comporte les réseaux à mécanisme interne et implicite du temps. Cette partie représente l'ensemble des réseaux de neurones récurrents. Dans le cas d'un modèle neuronal récurrent, la connectivité des unités dans les réseaux ne se limite pas à des connexions en avant (feedforward). Tout type de connexion est admis, c'est à dire d'un neurone à n'importe quel autre, y compris lui même. Cette architecture donne lieu à un comportement dynamique complexe. On distingue deux types de réseaux récurrents : les réseaux à récurrence globale et localement récurrent. La récurrence globale est représentée par la réinjection de la sortie du réseau ou de sa couche cachée dans la couche d'entrée. En revanche, la récurrence locale est représentée par un retour de l'activation d'un neurone à lui même. Dans les travaux de prédiction de séries temporelles effectués dans le cadre de cette thèse, on s'intéresse particulièrement à l'application des réseaux de neurones récurrents (RNR) pour résoudre le problème de prédiction des performances variables des composants qui constituent un système sur puce.

III.4 TDNN RESEAU DE NEURONE A DELAIS TEMPORELS

III.4.1 INTRODUCTION

Dans plusieurs applications le temps est une donnée indissociable des signaux d'entrée. La reconnaissance de la parole est un des domaines le plus étudié dans le cadre des réseaux neuronaux où le temps joue un rôle très important. Ce genre de problème nécessite un modèle capable de représenter des relations entre les différents signaux dans le temps.

Les réseaux de neurones, les plus adaptés pour de telles applications sont

ceux qui utilisent les connexions à délai(en anglais : "Time-Delay Neural Networks" TDNN) introduites par Alexander Waibel et Geoffrey Hinton [32]. Il s'agit, en fait, d'une utilisation particulière de connexions à poids partagés afin de propager l'information du temps à travers le réseau. Les applications liées au problème de la reconnaissance de l'écriture

sont par excellence des cas où la prise en compte du paramètre *temps* dans l'information est capitale afin d'espérer avoir des modèles capables de bonnes performances dans des tâches comme la classification de ces signaux

Pour illustrer ceci, nous allons présenter dans ce chapitre une architecture de réseaux de neurones conçue à cet effet, ses principes de base seront développés, nous énoncerons ses propriétés fondamentales, nous définirons par la suite l'unité temps retard, ainsi que la notion des poids partagés. En dernier lieu, nous aborderons l'architecture du TDNN proposée dans notre approche.

III.4.2 PROPRIETES DU TDNN

Le TDNN est un réseau de neurones multicouches qui peut être entraîné à reconnaître des structures spectrales spécifiques. La structure temporelle du TDNN est représentée au niveau d'abstraction croissante et la durée de la progression dans la couche d'entrée à la couche de sortie. Les trames d'entrée à temps-retardés permettent aux poids dans les couches initiales à rendre compte des variations dans la représentation spectrale du signal vocal.

Afin de réaliser une reconnaissance efficace le réseau neuronal à couche doit présenter les propriétés suivantes:

- avoir plusieurs couches et suffisamment d'interconnexions entre ses unités de calcul, afin d'assurer que le réseau va avoir la possibilité d'apprendre des surfaces de décision non linéaires assez complexes,
- être capable de représenter les relations entre les événements dans le temps, ces événements peuvent être des coefficients spectraux,
- l'utilisation de la notion de poids partagés dans le réseau.

III.4.3 TDNN ET APPRENTISSAGE

Un des points importants dans la mise en œuvre d'une solution à base de réseau de neurones concerne la méthode d'apprentissage retenue. Celle-ci doit permettre de converger rapidement vers le minimum global de la fonction de coût sélectionnée.

Le TDNN est capable d'effectuer une reconnaissance de bonne précision par un simple apprentissage du premier ordre, ce dernier qui est un apprentissage supervisé qui utilise la méthode de la rétropropagation.

L'algorithme de la rétro-propagation est une technique s'appliquant sur des neurones à activation continue (fonction sigmoïde). L'objectif de cet algorithme est de minimiser une fonction de coût "E", cette fonction est exprimée à partir de l'erreur quadratique, pour un

couple y_i et y'_i avec y_i la sortie calculée par le neurone d'indice i et y'_i la sortie désirée.

Cette erreur quadratique est donnée par la formule suivante :

$$E = 1/2 \sum_i^N (y'_i - y_i)^2 \quad (\text{III-16})$$

Où N est le nombre total de neurone de sortie.

Cette méthode se résume en deux passes (déjà cité dans le chapitre précédent)

- **Passé avant** : c'est une phase de calcul dans le sens direct, où chaque neurone effectue la somme pondérée de ses entrées et applique ensuite la fonction

d'activation F pour obtenir une sortie y_j donnée, par:

$$y_j = f(a_j);$$

$$a_j = \sum_i^N \sum_d^D W_{dij} * y_i(t-d) \quad (\text{III-17})$$

Où y_i est l'état du neurone de couche précédente et W_{ij} le poids de la connexion entre les deux neurones i et j . Cette phase dite de propagation permet de calculer la sortie du réseau en fonction de l'entrée.

- **Passé arrière** : On calcule l'erreur de ces résultats par rapport aux résultats attendus pour chaque neurone de sortie et on répercute cette erreur aux neurones des couches internes suivant des formules propres à l'algorithme du « Gradient-descend ».

La formule qui calcule l'erreur pour la couche de sortie est la suivante:

$$\delta_i = (\sigma'_i - \sigma_i) \sigma_i (1 - \sigma_i) \quad (\text{III-18})$$

$$f'(x) = \sigma_i (1 - \sigma_i) \quad (\text{III-19})$$

Où : σ'_i La sortie désirée et σ_i La sortie du neurone i .

Pour les neurones des couches inférieures :

$$\delta_i = \sigma_i (1 - \sigma_i) \sum_k W_{ik} * \delta_k \quad (\text{III-20})$$

Où: k est l'indice des neurones de la couche supérieure.

Après calcul de ces erreurs et les gradients associés, on ajuste les poids du réseau de neurones. Dans un TDNN la modification des poids est en fonction de la moyenne de l'erreur sur l'ensemble des neurones suivant la direction temporelle.

Soit w_{ij} le poids de la connexion reliant le i ème neurone de la couche inférieure et le j ème neurone de la couche supérieure, sa correction est donnée par :

$$\Delta W_{ij} = -h \left(\frac{1}{F} \right) \sum_d \frac{\partial E^d}{\partial w_{ij}^d} \quad (\text{III-21})$$

Où F est le nombre de neurones selon la direction temporelle de la couche supérieure, h est le pas d'apprentissage et $\frac{\partial E^d}{\partial w_{ij}^d}$ est le gradient de chaque neurone de la couche supérieure suivant la direction temporelle.

Il existe trois façons d'appliquer l'algorithme de la rétro-propagation :

Batch : mise à jour des poids après la représentation de toute la base d'apprentissage.

Séquentiel appelé aussi stochastique : mise à jour des poids après chaque exemple.

Incrémental : cette approche ajuste les paramètres à chaque intervalle de temps.

L'algorithme de la rétro-propagation est optimal mais représente quelques problèmes tels que ceux de la lenteur de convergences, d'arrêt prématuré de l'apprentissage ou de « sur apprentissage » [2], dans la plupart des cas le principe consiste à apporter quelques modifications à l'algorithme afin de résoudre ces problèmes. Cette possibilité de modifier concerne quelques caractéristiques liées à l'apprentissage : le gain d'adaptation et la présentation des exemples de la base d'apprentissage.

- **Le gain d'adaptation** : le choix des paramètres d'un algorithme d'apprentissage influe beaucoup sur la rapidité de calculs. Le principe de descente du gradient consiste à effectuer de manière itérative (pas par pas) une modification des poids jusqu'à arriver à un minimum sur la fonction du coût ; si ce pas (pas d'apprentissage ou appelé aussi gain d'adaptation) est défini trop petit ; le nombre de pas nécessaire pour la convergence peut s'avérer relativement important et donc il ralentit d'une manière non négligeable l'apprentissage, à l'inverse, si le pas est trop important, l'algorithme peut devenir instable. Une méthode consiste à modifier l'algorithme d'ajouter à ce gain d'adaptation un terme de filtrage appelé « momentum ».
- **Présentation des exemples** : dans une méthode globale, les poids sont modifiés après avoir calculé l'erreur sur tout le corpus d'apprentissage. Lorsque les poids sont modifiés après chaque présentation d'un exemple, on parle de gradient adaptatif ou stochastique. Cet algorithme est plus rapide et permet d'atteindre avec une probabilité plus grande les bons optimums lorsque la base d'apprentissage est importante.

III.4.4L'UNITE TEMPS-RETARD

L'unité de base, utilisée dans de nombreux réseaux neuronaux, calcule la somme pondérée de ses entrées avant de lui appliquer une fonction non linéaire f :

$$y_j = f\left(\sum_{i=1}^N W_{ij} \bullet x_i\right) \quad (\text{III-22})$$

On suppose que l'unité j est connectée aux N unités de la couche précédente.

Dans le TDNN, cette unité de base est modifiée en lui ajoutant une dimension temporelle par introduction de la notion de retard, figure III.8.

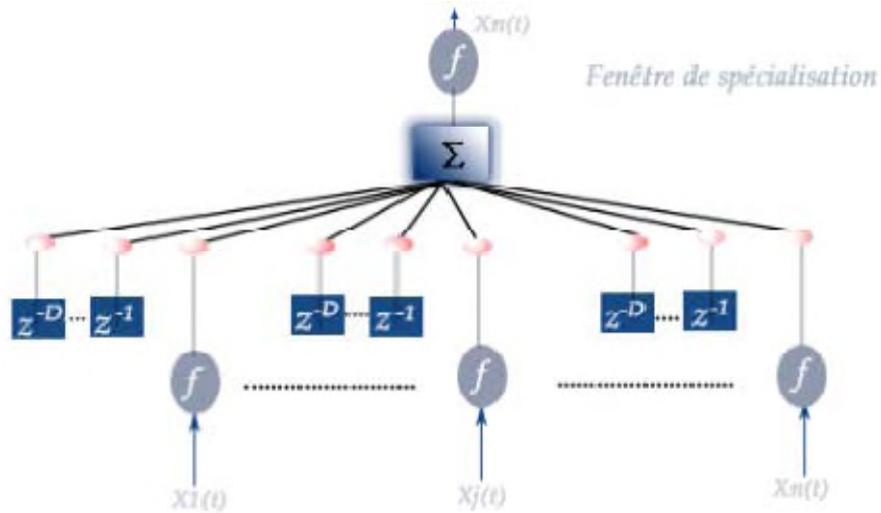


Figure III-8: Chaque cellule du TDNN comprend une structure d'entrée qui teste les valeurs du vecteur de données courantes $x_i(t)$ directement, ainsi que D vecteurs retardés $x_i(t-D)$. z^{-D} indique la quantité du temps retard dans l'entrée.

Les entrées à l'unité vont être maintenant multipliées par plusieurs poids, un pour chaque retard et un pour l'entrée non retardée.

$$y_j(t) = f\left(\sum_{i=1}^N \sum_{d=0}^D W_{ij}^d \cdot x_i(t-d)\right) \quad (\text{III-23})$$

L'équation précédente montre une manière de déplier le temps dans l'espace, cela permet de traiter statiquement l'architecture dynamique des réseaux TDNN avec deux contraintes sur la matrice des poids:

- elle n'est pas entièrement connectée : $y_j(t)$ ne dépend pas des unités avant $t-D$ ou après t .
- certains poids sont partagés, par exemple, les unités calculant $y_j(t)$ et $y_j(t+1)$ partagent entièrement leur poids.

III.4.4.1 Les fenêtres temporelles

Le concept de fenêtre temporelle implique que chaque neurone de la couche $l+1$ n'est connecté qu'à un sous ensemble de la couche l (nous n'avons plus une connectivité totale) . La taille de cette fenêtre est la même entre deux couches données. Cette fenêtre temporelle permet que chaque neurone n'ait qu'une vision locale du signal, il peut être vu comme une unité de détection d'une caractéristique locale du signal.

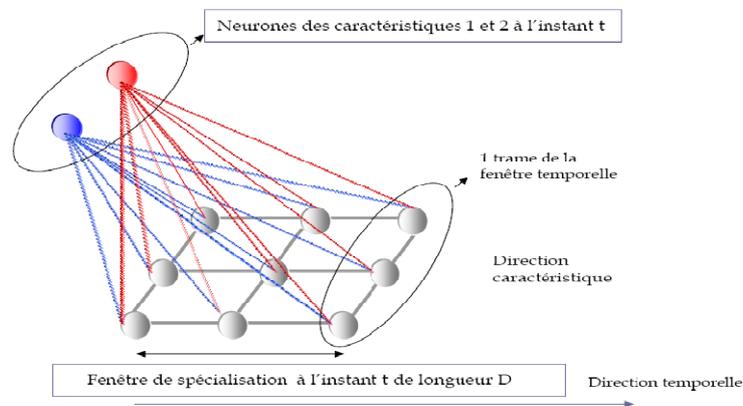


FIGURE III-9 CONNECTIVITE LOCALE ENTRE LES NEURONES DE LA COUCHE $l+1$ A L'INSTANT t AVEC LES NEURONES DE LA FENETRE DE SPECIALISATION A L'INSTANT t

III.4.5 PRINCIPE DES POIDS PARTAGES

Waibel a proposé un réseau MLP (Multi Layer perceptron) avec des poids partagés afin de mieux cerner l'aspect dynamique du signal. Cela revient à contraindre le réseau à avoir des poids similaires pour plusieurs unités.

Le comportement présumé du cerveau humain utilise le concept des poids partagés, en effet, dans le cerveau humain des neurones détectent certains traits dans des petites régions de la rétine essentiellement de la même manière dans toutes ces régions, ce qui fait qu'on a plusieurs neurones qui calculent la même fonction sur des entrées différentes.

Conséquence directe de cette contrainte, le réseau est de moindre dimension sans pour autant perdre de son pouvoir discriminant, et même qu'il en résulte souvent un pouvoir de généralisation plus accru par la suite, car le niveau d'abstraction est plus élevé. Ces réseaux permettent de mieux prendre en compte l'aspect temporel du signal.

Le TDNN code un contexte dont les potentielles d'action parviennent de plusieurs trames aux instants t , $t+1$, $t+2$, ... Selon la taille du contexte pris en compte à la différence d'un MLP, les connexions entre les couches d'un TDNN sont partielles, les matrices de poids sont réduites à l'identité le long de l'axe temporel, afin d'apprendre les caractéristiques du signal et leurs relations temporelles indépendamment de la position dans le temps.

Donc ces réseaux sont théoriquement mieux adaptés à la nature dynamique de l'écriture que des MLPs simples.

III.4.6 ARCHITECTURE GENERALE DU TDNN

Généralement, un TDNN est construit sur la base d'un MLP ayant une couche d'entrée, une couche de sortie et une ou plusieurs couches cachées

Les neurones de la première couche cachée sont reliés aux neurones de la couche d'entrée par des connexions à retard, et les neurones de la deuxième couche cachée sont connectés à ceux de la première couche cachée par le même principe. Ce nombre de retards ou pas de temps définit la largeur de la "fenêtre de spécialisation" appelée aussi fenêtre glissante représenté par la figure III.9.

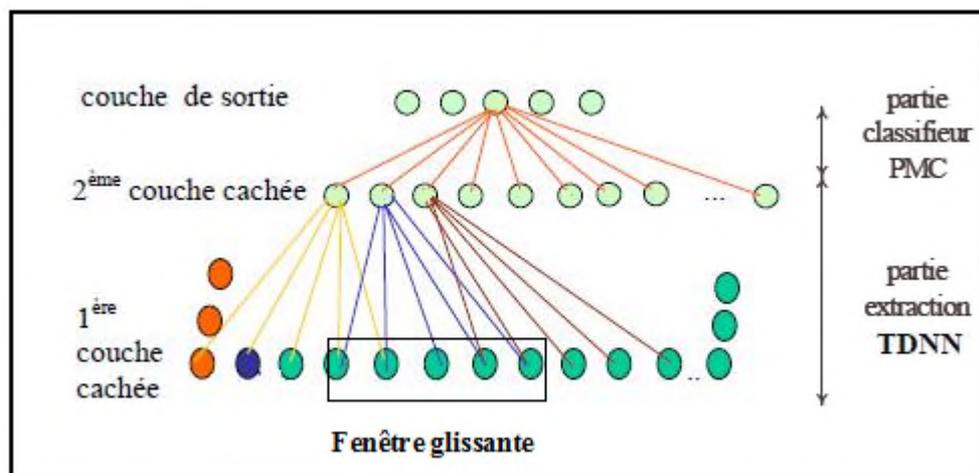


Figure III-9: Représentation du TDNN

Les liaisons comprises entre la couche d'entrée du réseau et la première couche cachée comportent des délais, cette notion de retard permet de modifier l'unité de base en lui ajoutant une dimension temporelle. Les entrées à l'unité vont être maintenant multipliées par plusieurs poids, un pour chaque retard et un pour l'entrée non retardée afin de permettre la mémorisation de l'information passée, de même pour les liaisons entre la première et la deuxième couche cachée cette partie du réseau forme la partie extraction.

Enfin, la deuxième couche cachée et la couche de sortie compose la partie classification (MLP entièrement connecté).

La première couche du réseau acquiert les caractéristiques du signal. Une ou plusieurs couches cachées du réseau de neurones (phase d'extraction) transforment une séquence de vecteurs caractéristiques en une autre séquence de vecteurs caractéristiques d'ordre supérieur. Le champ de vision du neurone est restreint à une fenêtre temporelle limitée. Avec la contrainte des poids partagés, le même neurone est dupliqué dans la direction temps (soit la même matrice des poids dupliquée) pour détecter la présence ou l'absence de la même caractéristique à différentes places le long de la trajectoire du signal. En utilisant plusieurs neurones (nombre de pas) à chaque position temporelle, le réseau de neurones effectue la détection de caractéristiques différentes : les sorties des différents neurones produisent un nouveau vecteur caractéristique pour la couche supérieure. La composante temporelle de la représentation du signal d'origine est éliminée au fur et à mesure. Pour compenser cette perte d'informations, le nombre de caractéristiques est multiplié. Nous avons donc une architecture qui convertit progressivement les informations temporelles en informations caractéristiques.

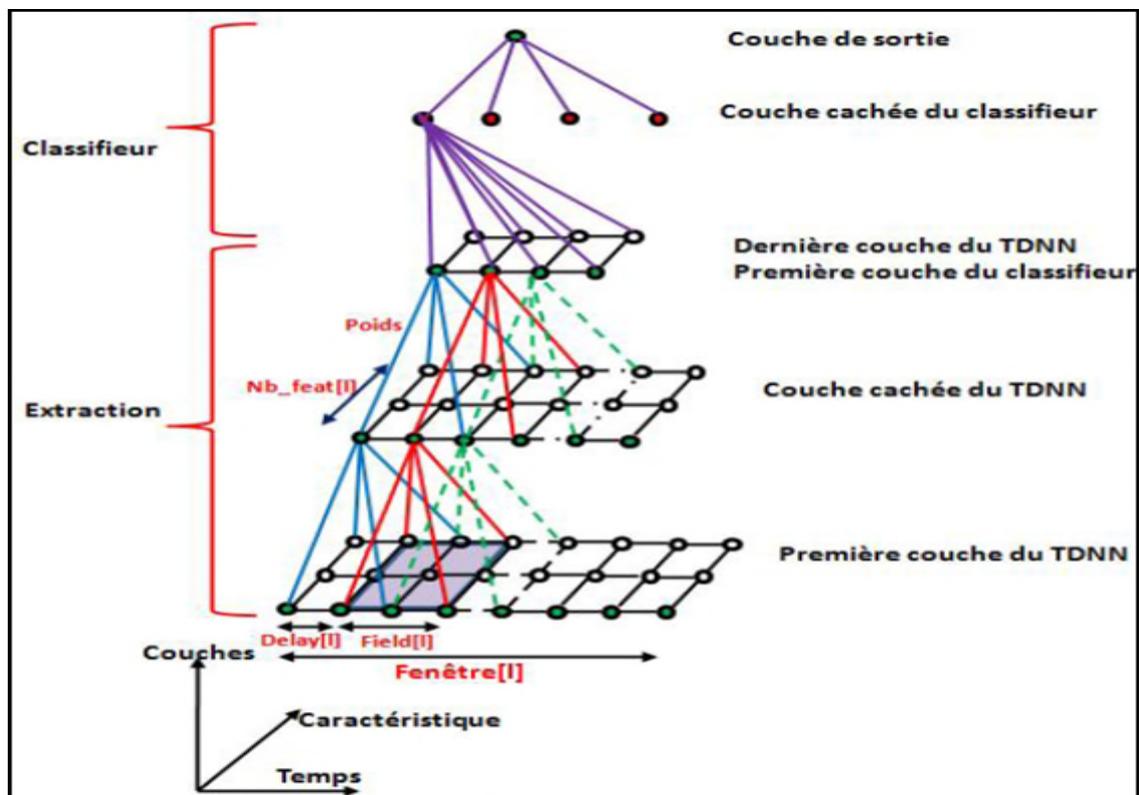


Figure III-10: Architecture du TDNN pour l'écriture

La figure III.10ci dessus représente l'architecture du TDNN appliqué pour la reconnaissance de l'écriture manuscrite non contrainte, [3].

Nous nous sommes inspirés de cette architecture pour la réalisation de l'architecture proposée pour notre reconnaissance.

Cette architecture comprend les deux parties suivantes :

- une partie extraction permettant de transformer progressivement les caractéristiques en grandeurs de plus en plus significatives vis-à-vis du problème.
- la seconde correspond à un MLP, il reçoit en entrée l'ensemble des sorties de la partie TDNN.

Ces deux blocs sont complètement paramétrables, ils sont décrits par les grandeurs présentées ci-après.

La partie extraction se caractérise par :

- le nombre de couches, (chaque couche comprend deux directions : une direction temporelle et une direction caractéristique), nb_layer
- le nombre de neurones de chaque couche selon la direction temporelle, $window_T$.
- le nombre de neurones de chaque couche selon la direction caractéristique, nb_feat .
- la taille de la fenêtre temporelle vue par chaque couche (sauf celle d'entrée) soit le nombre de neurones de la couche vue par un neurone de la couche $l + 1$, $field_T$
- le délai temporel (nombre de neurones) entre chaque fenêtre, $delay$.

Un neurone est identifié de la partie TDNN par sa couche l , sa caractéristique f et son emplacement temporel. Chaque neurone est caractérisé par :

- une sortie, ou encore activation du neurone, $x[l][f][t]$
- une matrice des poids des entrées, $x[l-1][f][f[l-1]] [t]$
- un vecteur de poids des biais, $w_biais [l-1][f]$
- la somme pondérée des entrées, $v[l][f][t]$
- le terme d'erreur pour la rétro propagation du gradient, $y[l][f][t]$
- le gradient, $delta[l-1][f][f[l-1]] [t]$
- une fonction d'activation d type tangente hyperbolique

La partie Classification se caractérise par :

- le nombre de couches, NN_nb_layer
- Le nombre de neurones de chaque couche NN_nb_neuron

Un neurone de la partie classifieur est identifié par sa couche l , et son emplacement t . Chaque neurone de cette partie est défini par :

- une sortie, ou encore activation du neurone, $x[l][t]$
- une matrice des poids des entrées, $w[l-1][t][t[l-1]]$
- un vecteur de poids des biais, $w_biais[l-1]$
- la somme pondérée des entrées, $v[l][t]$
- le terme d'erreur pour la rétro propagation du gradient, $y[l][t]$
- le gradient, $\delta[l-1][t][t[l-1]]$
- une fonction tangente hyperbolique pour les couches cachées,
- une fonction d'activation de type Softmax pour la couche de sortie.

III.5 Conclusion

Dans la recherche de meilleurs algorithmes capables de résoudre une classe de données de problèmes, une tendance actuelle veut que l'on bâtit des méthodes complexes incorporant des connaissances et des techniques issues de plusieurs horizons.

Si l'on est exigeant sur la qualité de la solution, et que l'on souhaite un fonctionnement automatique, des critères pertinents sont traités et améliorés.

Le choix de l'approche de classification à développer est un autre défi et se justifie par l'utilisation des modèles neuronaux de connaissances, tels que les réseaux de neurones TDNN, qui nous a permis de traiter efficacement l'ensemble des caractères isolés en gardant leurs informations spatiales et l'ordre d'écriture de chaque caractère du manuscrit arabe.

L'amélioration des performances avec la combinaison des algorithmes d'apprentissage .

-

CHAPITRE IV:IMPLEMENTATION, RESULTATS ET DISCUSSION

IV.1 INTRODUCTION

Dans ce chapitre, nous nous pencherons sur le système que nous avons implémenté, basé sur les réseaux de neurones à convolution (présenté précédemment). L'architecture TDNN conçue est appliquée à la reconnaissance des caractères arabes en ligne.

Passons maintenant à la reconnaissance des caractères arabes isolés, ou nous verrons en détails le système implémenté basé sur les réseaux de neurones à convolution. Nous nous préoccupons dans un premier lieu à concevoir la base de données que nous avons baptisé NOUN DATABASE dans sa première version, conçue à l'aide d'une acquisition en ligne d'un nombre de caractères pour la validation de notre système de reconnaissance.

IV.2 PRESENTATION D'UN RECONNAISSEUR DE CARACTERES ISOLES

D'une façon générale, un système de reconnaissance de caractères acquiert les données provenant d'un équipement en ligne (tablette dans notre cas) ou hors ligne (scanner par exemple), en assure le traitement. Le schéma général d'un système pareil comporte trois étapes principales (voir figure IV-1). La première étape consiste à faire le prétraitement (normalisation). La seconde étape est l'extraction des caractéristiques, ces dernières sont utilisées par un composant de classification (dernière étape) pour attribuer une étiquette à la forme d'entrée.

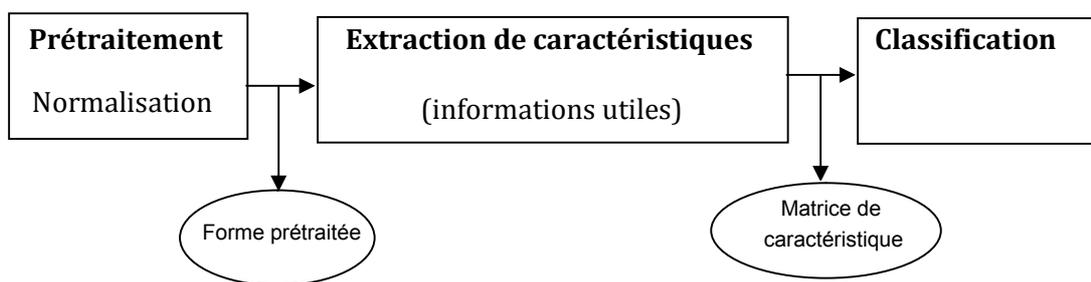


Figure IV-1: Schéma général d'un SRC

Le système décrit ci-dessus reçoit en entrée un signal en ligne, ce dernier est échantillonné en un nombre de points fixe puis normalisé. A chaque point correspond une séquence de "n"

caractéristiques, on aura une matrice de dimension: (nombre de points * nombre de caractéristiques), qui sera présentée à un réseau de neurone à délai temporel (le TDNN).

En paramétrant le bloc TDNN, nous avons cherché le meilleur compromis performance architecture, les spécifications suivantes sont respectées:

- Les caractères sont acquis à l'aide d'une tablette Wacom
- L'acquisition du caractère est accompagnée de quelques renseignements sur le scripteur.
- L'acquisition concerne les 28 lettres arabes écrites sous leur forme isolée.
- Aucune analyse syntaxique ou sémantique n'est considérée.

IV.3 CONSTRUCTION DE LA BASE DE DONNEES NOUN:

Comme décrit dans le chapitre 2 (regroupant les principales bases de données), la base de données ADAB est la seule base de données à caractère en-ligne validée et utilisée regroupant les lettres et mots arabes. Vu son indisponibilité et son accès payant, nous avons pensé à construire notre propre base de données baptisée NOUN DATABASE qui sera notre fer de lance pour les éventuels projets de recherches dans la reconnaissance du manuscrit arabe.

IV.3.1 DISPOSITIFS MATERIELS

IV.3.1.1 Ordinateur

Pour la réalisation des expériences, un ordinateur de bureau a été utilisé avec :

- Un processeur Pentium Core2Duo avec vitesse d'horloge 2.0 GHz
- Mémoire vive RAM installée 2 Go.

Le système d'exploitation utilisé est le Windows Vista (32 bits) édition familiale.

Toutes les expériences sont développées sous MATLAB version 7.8.0.347 (R2009a).

IV.3.1.2 Tablette PC

Pour la construction de la base de données « NOUN », une tablette WACOM BAMBOO version 5.08-6 a été exploitée et illustrée dans la figure suivante :

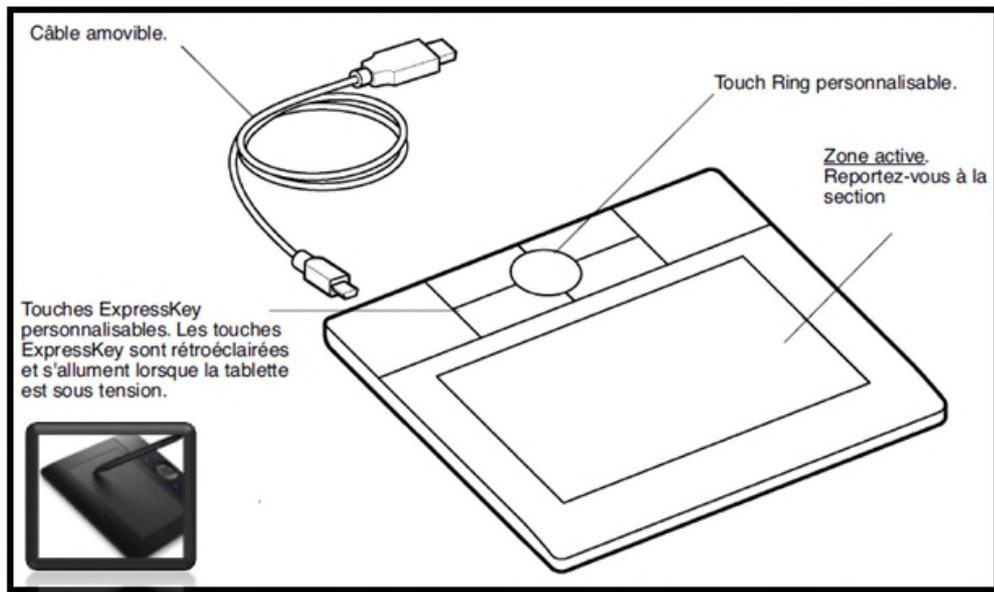


Figure IV-2: Fonctionnalités de la tablette WACOM

La tablette dispose d'un stylet sans fil et sans pile, contribue à réaliser le travail en activant ses fonctions d'encre numérique. Le stylet est un outil à main levée sensible à la pression. Il permet d'effectuer quatre opérations de base : le pointage, le clic, le double-clic et le déplacement.

La figure IV.3 montre les composants de ce stylet et le réglage de paramètres.

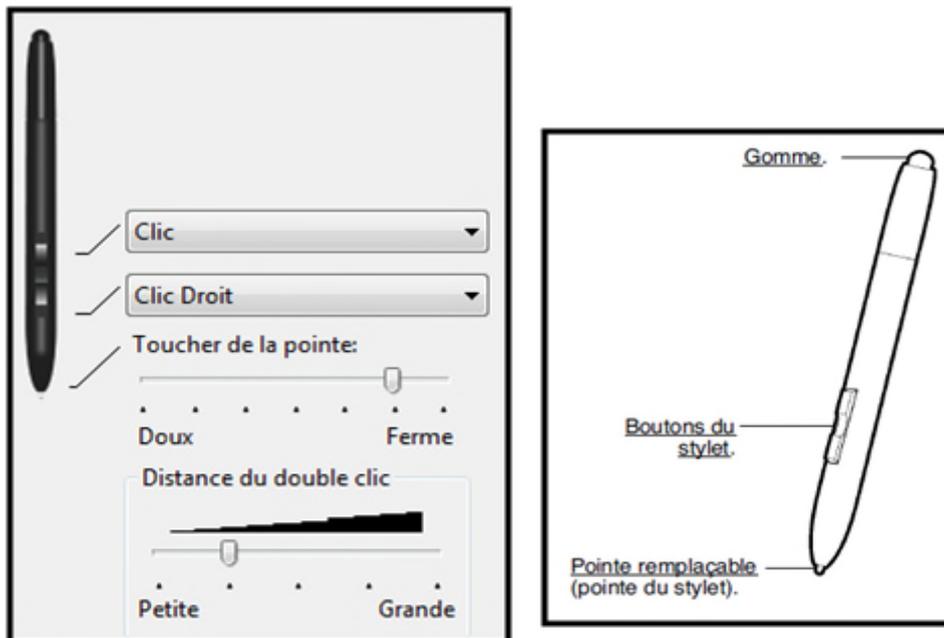


Figure IV-3: Composants de stylets et réglage de paramètres

Il faut tenir le stylet comme un crayon classique. Le stylet peut être incliné pour travailler.

Les boutons de stylet sont positionnés à pouvoir appuyer facilement dessus avec la pouce ou l'index. (Voir figure IV.4)

Avant l'utilisation de la tablette, il faut organiser l'espace de travail de manière à travailler confortablement. Il faut placer la tablette, le stylet et le clavier de façon qu'ils soient aisément accessibles. Il faut aussi placer le moniteur de manière à le voir avec un minimum de fatigue oculaire. Pour une utilisation optimale, la tablette doit être orientée de sorte que le curseur de l'écran se déplace dans le même sens de la main du scripteur. (Voir la figure IV.5)

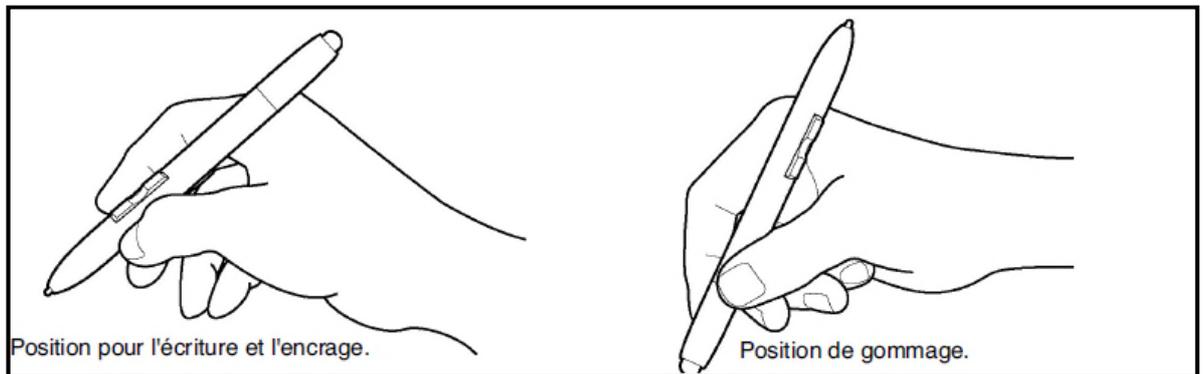


Figure IV-4: Positionnement de stylet

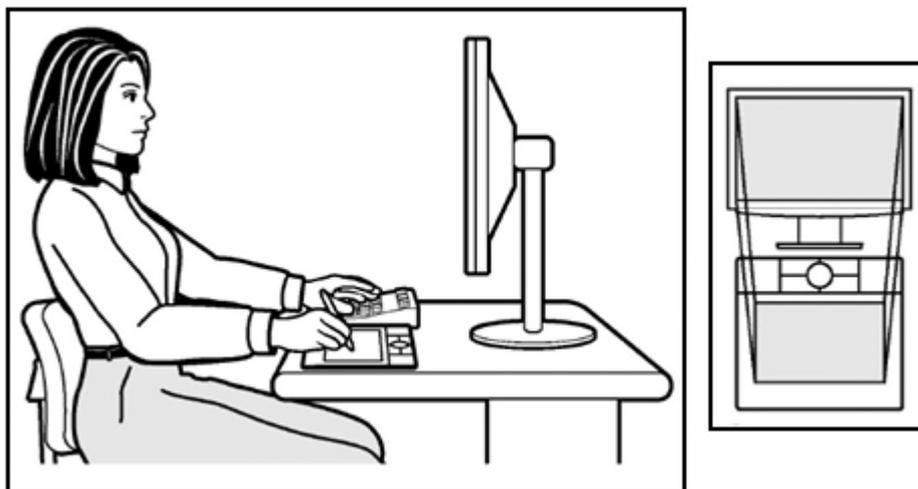


Figure IV-5: Zone de travail sur tablette

IV.3.2 FICHE D'ECRITURE SCRIPTEUR

IV.3.2.1 Alphabet

L'alphabet Arabe est composé de 28 lettres avec leurs formes isolées. Le scripteur est invité à saisir tout l'alphabet un nombre N d'occurrences. Il doit respecter la fiche mentionnée sur la figure suivante :

خ	ح	ج	ث	ت	ب	ا
Kha	Ha	Jim	Tha	Ta	Ba	Alef
ع	ح	ج	ث	ت	ب	ا
ص	ش	س	ر	ر	د	د
Sad	Chin	Sin	Zain	Ra	Dhal	Dal
ص	ش	س	ز	ر	ذ	د
ق	ف	ع	ع	ظ	ظ	هـ
Qaf	Fa	Gham	Aim	Thah	Ttah	Dhad
ق	ف	ع	ع	ظ	ظ	هـ
ي	هـ	و	ن	م	ل	ك
Ya	He	Waw	Noun	Mim	Lam	Kaf
ي	هـ	و	ن	م	ل	ك

Figure IV-6: Alphabet saisie par le scripteur

IV.3.2.2 Interface d'acquisition

L'interface de construction des échantillons de caractères arabes isolés est la suivante :

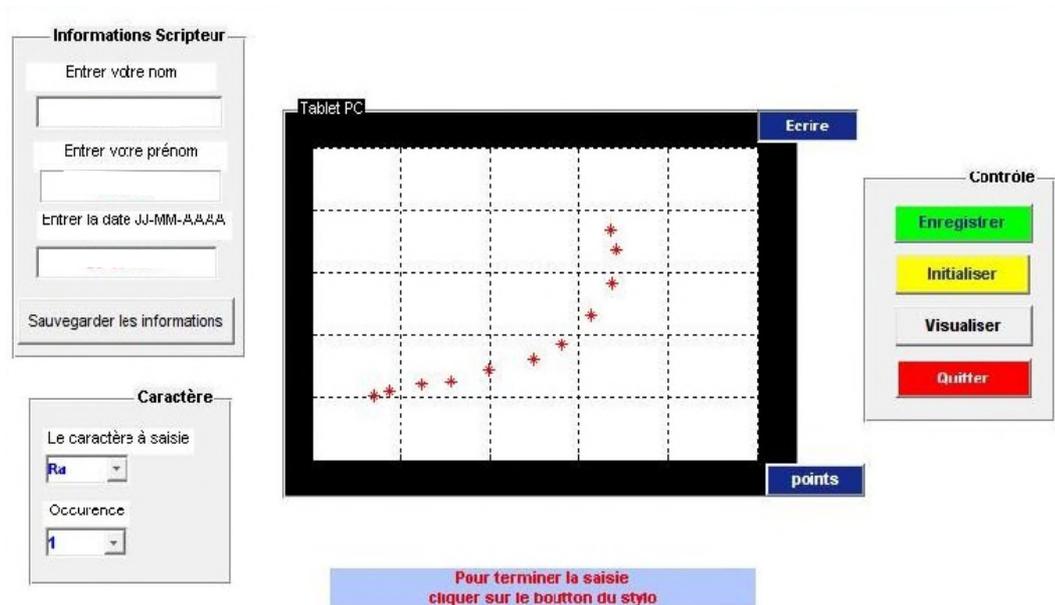


Figure IV-7: Interface d'acquisition

Le scripteur doit inscrire son *nom*, *prénom* et la *date de saisie* dans l'onglet « **Information Scripteur** » mentionné à gauche dans l'interface.

En bleu, les deux boutons : « *Écrire* » et « *points* » sont réservés pour saisir le caractère et les points des caractères respectivement.

Le scripteur doit faire une sélection du caractère à saisir et quelle occurrence sans l'onglet « **caractère** ».

L'onglet « **contrôle** » à droite sert à appliquer quelques contrôles utiles pour la construction des échantillons. En vert, le bouton « *enregistrer* » pour la sauvegarde, en jaune le bouton « *initialiser* » pour la remise à zéro de toutes les informations sur l'interface. En gris pour la visualisation du caractère sauvegardé et en rouge pour quitter l'interface.

IV.3.3 DESCRIPTION DE LA BASE

La base de données NOUN dans sa première version est développée pour initier la recherche et le développement des systèmes de reconnaissance du manuscrit arabe en ligne. Elle contient 2800 caractères isolés, en arabe. Elle a été collectée auprès d'environ 20 scripteurs différents, chacun a écrit les lettres de l'alphabet arabe cinq fois. Cette base

de données a été développée entre : le laboratoire SIMPA (Signal Image Parole) à l'Université des Sciences et de la Technologie d'Oran USTO-MB et le laboratoire LaRATIC (Laboratoire de Recherche Appliquée et des TICs) à l'Institut National des Télécommunications et des Technologies d'Information et de la Communication INTTIC.

IV.4 EXPERIMENTATION ET RESULTATS:

IV.4.1 PRETRAITEMENT DES CARACTERES :

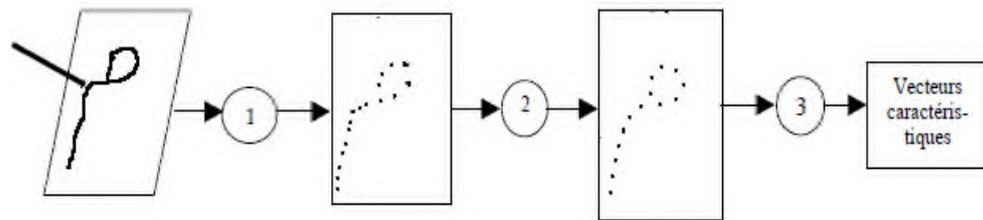


Figure IV-8: Représentation du processus d'extraction en ligne

(1 : Acquisition du signal – 2 : Échantillonnage – 3 : Extraction des caractéristiques)

Lors de la construction de notre base de donnée nous avons fait appel à la tablette WACOM, les données brutes du signal d'écriture contiennent d'importantes variations selon la vitesse d'écriture se traduisant par un nombre différent des point formant la lettre. Entraîner un réseau de neurones avec de telles données peut engendrer de très faibles performances pouvant atteindre les 70% de taux d'erreurs. L'échantillonnage spatial améliore sensiblement le taux de performance.

IV.4.1.1 Échantillonnage spatial:

L'étape de l'échantillonnage spatial est une étape très importante qui permet d'éliminer l'information redondante symbolisée par la répétition de points en conservant l'information utile du signal. En effet, le signal d'écriture varie et cela pour le même scripteur, le temps de formation d'un caractère ainsi que sa forme peuvent varier sensiblement d'un scripteur à un autre, et d'une occurrence à une autre.

L'échantillonnage spatial consiste à représenter une séquence de points dont les coordonnées $[X(n), Y(n)]$ (n étant le numéro du point), espacés par une distance identique selon la longueur du trait en fixant le nombre de point. Par expérimentation, nous avons fixé le nombre de points à 17 pour chaque caractère.

```

\ Définition du trait: une séquence de points entre un "poser" et un "lever" du stylet
N: nombre de points d'échantillonnage
Réservation de l'espace mémoire pour les vecteurs suivants: X, Y, PenUpDown
\ Calcul de la longueur totale du signal
  Si un seul trait
    Longueur = Longueur + distance entre les points
  Si plusieurs traits
    Longueur = Longueur + Longueur entre les traits
\ Calcul de la distance d'échantillonnage
Dist-echa = Longueur \ (N-1)
Détermination des N-1 points par interpolation

```

Figure IV-9: Algorithme d'échantillonnage spatial

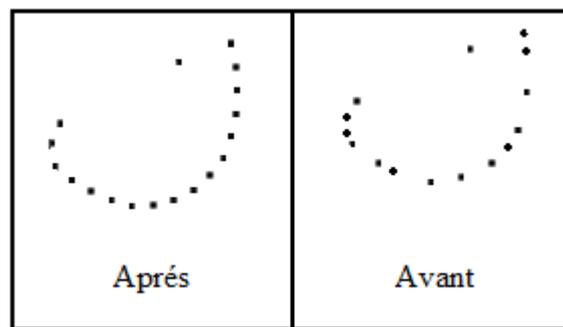


Figure IV-10: Caractère "Noun" avant et après échantillonnage spatial

IV.4.1.2 Extraction des caractéristiques:

C'est l'étape la plus pertinente de cette phase qui consiste à récupérer l'information utile et pertinente du signal acquis. Néanmoins cette opération est précédée par une phase de normalisation effectuant le recentrage du caractère afin d'obtenir une représentation invariante aux translations et aux distorsions spatiales. Le caractère est recentré par rapport au centre (x_0, y_0) (comme montré dans la figure IV-12) puis normaliser par rapport à la taille maximale du caractère.

Une fois la normalisation effectuée, on passe à la partie extraction qui consiste à prélever de chaque point du caractère un ensemble d'informations : les coordonnées en x et y, la direction du mouvement, la courbure de la trajectoire ainsi que la position du poser\lever du stylet.

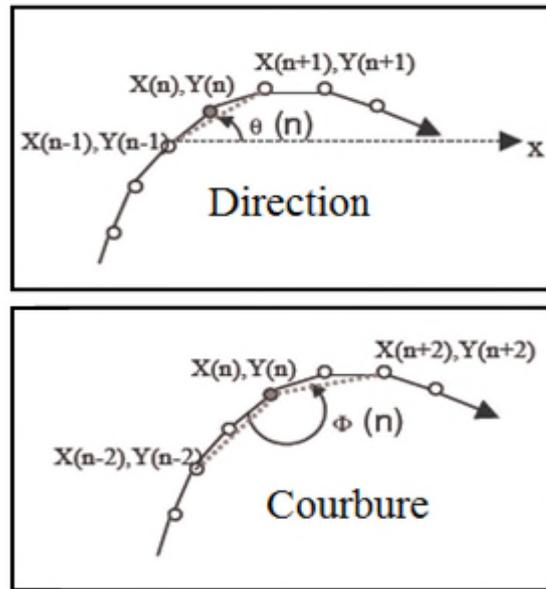


Figure IV-11: illustration des angles de la direction (θ) et de courbure (Φ)

L'algorithme est donné comme suit:

```

Allocation de la matrice en sortie X (matrice des caractéristiques)
\ \ Calcul du centre du caractère [x0,y0]
Initialisation xmax=x(0), xmin=x(0), ymax=y(0), ymin=y(0)
Balayage sur les points "n" de 0 jusqu'au N-1
  Si x(n) > xmax alors xmax =x(n)
  Si x(n) < xmin alors xmin =x(n)
  Si y(n) > ymax alors ymax =y(n)
  Si y(n) < ymin alors ymin =y(n)
x0=(xmin+xmax) \2
y0=(ymin+ymax) \2
\ \ Calcul du facteur d'échelle Δ
Δ_y=ymax-ymin
Δ_x=xmax-xmin
Si Δ_x < Δ_y alors Δ= Δ_y
  Sinon Δ= Δ_x
\ \ Calcul des nouvelles coordonnées x[n][0] et x[n][1]
Balayage des points de (de X et de Y)
x[n][1]= (X[n]-x0)\ Δ
x[n][2]= (Y[n]-y0)\ Δ
x[n][7]= PenUpDown[n]
\ \ Calcul de la direction x[n][3] et x[n][4] (cos et sin)
Balayage des points
  \ \ Calcul de la longueur de la corde dS
  dx=X[i+1]-X[i-1]
  dy=Y[i+1]-Y[i-1]
  dS=sqrt(dx*dx+dy*dy);
  if (dS==0) alors x[i][3]=0
    x[i][4]=0
  else x[i][3]=dx/dS
    x[i][4]=dy/dS
\ \ Calcul de la courbure x[n][5] et x[n][6]
Balayage des points
x[n][5]= x [i+1][3]* x [i-1][3]+ x [i+1][4]* x [i-1][4]
x[n][6]= x [i+1][3]* x [i-1][4] - x [i+1][4]* x [i-1][3]

```

Figure IV-12: Algorithme de normalisation et d'extraction des caractéristiques

Alors pour résumer, l'extraction des caractéristiques de chaque point du caractère nous permet de construire une matrice de 7×17 , les caractéristiques étant: les coordonnées en x, les coordonnées en y, les cosinus directeurs de la direction ($\cos \theta$ et $\sin \theta$), les cosinus directeurs de la courbure ($\cos \Phi$ et $\sin \Phi$) et la position poser\lever du stylet. C'est la matrice que nous propagerons dans notre architecture neuronale TDNN (Figure IV-13).

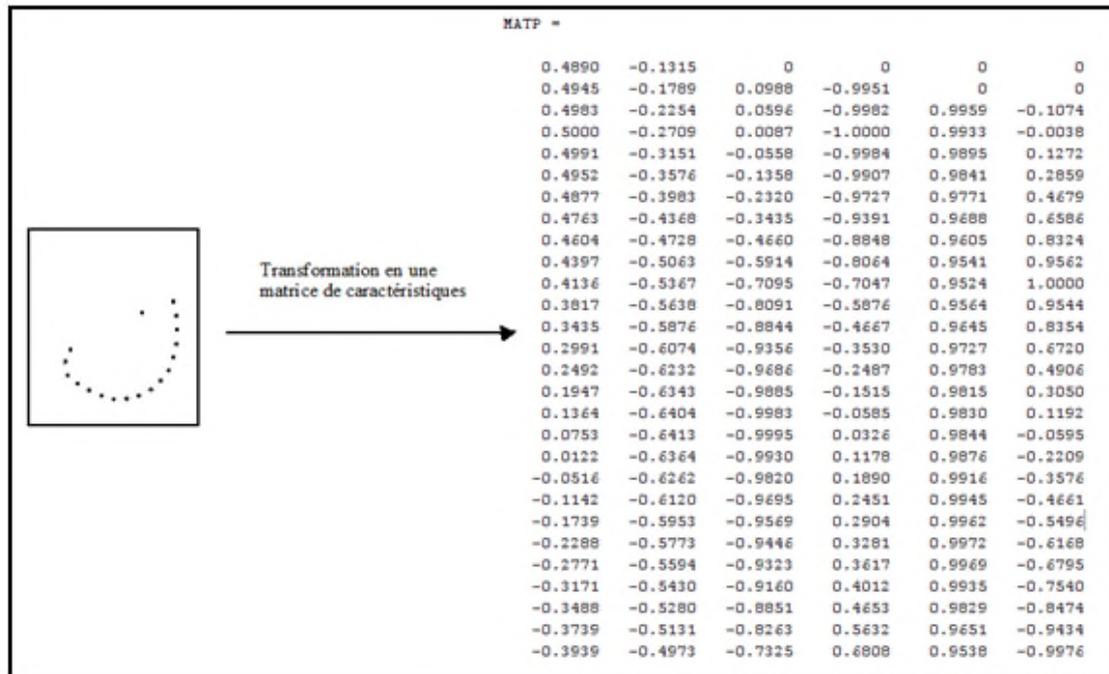


Figure IV-13: L'extraction des caractéristiques de la lettre Noun.

L'apprentissage, c'est-à-dire la détermination des poids est un des points importants dans la mise en œuvre d'un système neuronal est la méthode d'apprentissage, ces calculs utilisent l'ensemble des données de la série temporelle pour chercher les meilleurs poids afin que le réseau de neurones reproduise le comportement du système. Idéalement, ceux-ci doivent permettre de converger rapidement vers le minimum global de la fonction de coût sélectionnée. Le TDNN est un enchaînement de deux sous réseaux, le premier est un réseau dynamique et le deuxième est un réseau statique (comme montré précédemment). Pour l'apprentissage de ces deux sous réseaux, on peut appliquer la méthode d'apprentissage tel que la méthode de rétropropagation du gradient avec la contrainte de poids partagés. Toutefois, la version utilisée est la version dite du "gradient stochastique" qui permet de converger souvent plus rapidement que le gradient vrai, ou bien les méthodes de deuxième ordre tel que l'algorithme de Lavenberg-Marquardt, l'algorithme Newton, Quasi-Newton, Gauss-Newton. Nous avons testé des méthodes du premier ordre

qui y sont la rétropropagation du gradient (pour des raisons de calcul et de l'espace mémoire) ainsi que rétropropagation du gradient "résistante" : Rprop et deux méthodes du second ordre, celle de *Lavenberg-Marquardt* (LM) et du gradient conjugué (SCG).

IV.5 TOPOLOGIE ET PARAMETRES

IV.5.1 ARCHITECTURE REQUISE DU TDNN(DANS UN PREMIER TEMPS):

La propriété principale du TDNN est la capacité d'identifier les caractéristiques locales indépendamment de leur position dans le temps. Le TDNN est un réseau *feedforward* dynamique, où l'information se propage de l'entrée vers la sortie sans retour en arrière et la dynamique se situe dans la couche d'entrée, sous forme de retard. Le TDNN se singularise d'un réseau de neurones classique, tel que le réseau multicouche (PMC) par le fait qu'il prenne en compte une certaine notion du temps. C'est à dire qu'au lieu de prendre en compte tous les neurones de la couche d'entrée en même temps, il prend une fenêtre du spectre puis effectue un balayage temporel. Ce qui permet au réseau de prendre en considération les caractéristiques locales de la série temporelle de précipitations. Notons que le TDNN utilise la contrainte de poids partagés ce qui lui permet de réduire le nombre de paramètres du réseau neuronal en induisant ainsi une capacité de généralisation plus importante.

L'architecture de base testée du TDNN est un réseau à 3 couches, la première étant la couche d'entrée du réseau, la seconde est la couche cachée de la partie extraction ainsi que la première couche de la partie classification (la fonction d'activation étant une sigmoïde), la dernière est la couche de sortie (la fonction d'activation étant une fonction linéaire).

La couche d'entrée regroupe 119 neurones (17*7 caractéristiques) balayée par une fenêtre de convolution qui offre une vision locale de 4 neurones, le délai entre les fenêtres est estimé à 3 neurones selon l'axe temporel. La couche cachée de la partie extraction possède 8 neurones selon l'axe temporel et 14 neurones selon l'axe caractéristique.

Cette couche est l'entrée de la partie PMC, à la sortie on aura un nombre de neurones correspondant à la classification désirée: 28 neurones (par rapport au nombre des caractères du manuscrit arabe). Voir figure IV.15

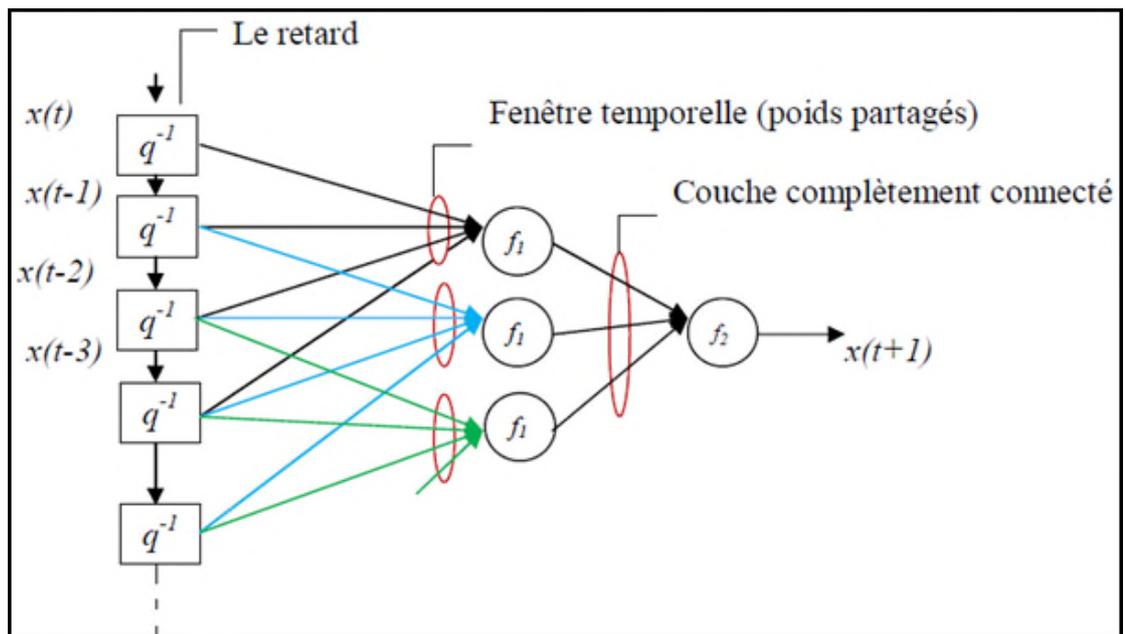


Figure IV-14: TDNN avec une fenêtre temporelle de taille 4 reliée à chaque neurones de la couche caché.

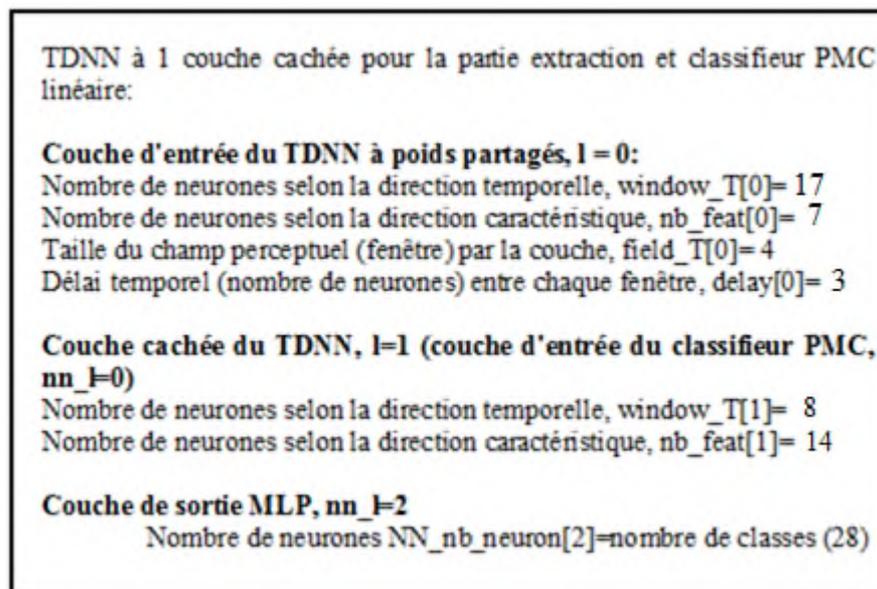


Figure IV-15: TDNN à 1 couche cachée

IV.5.2 APERÇU SUR LE MODELE

Dans la figure IV-16, il est bien montré l'aperçu du système exécuté en matlab7 pour notre modèle TDNN.

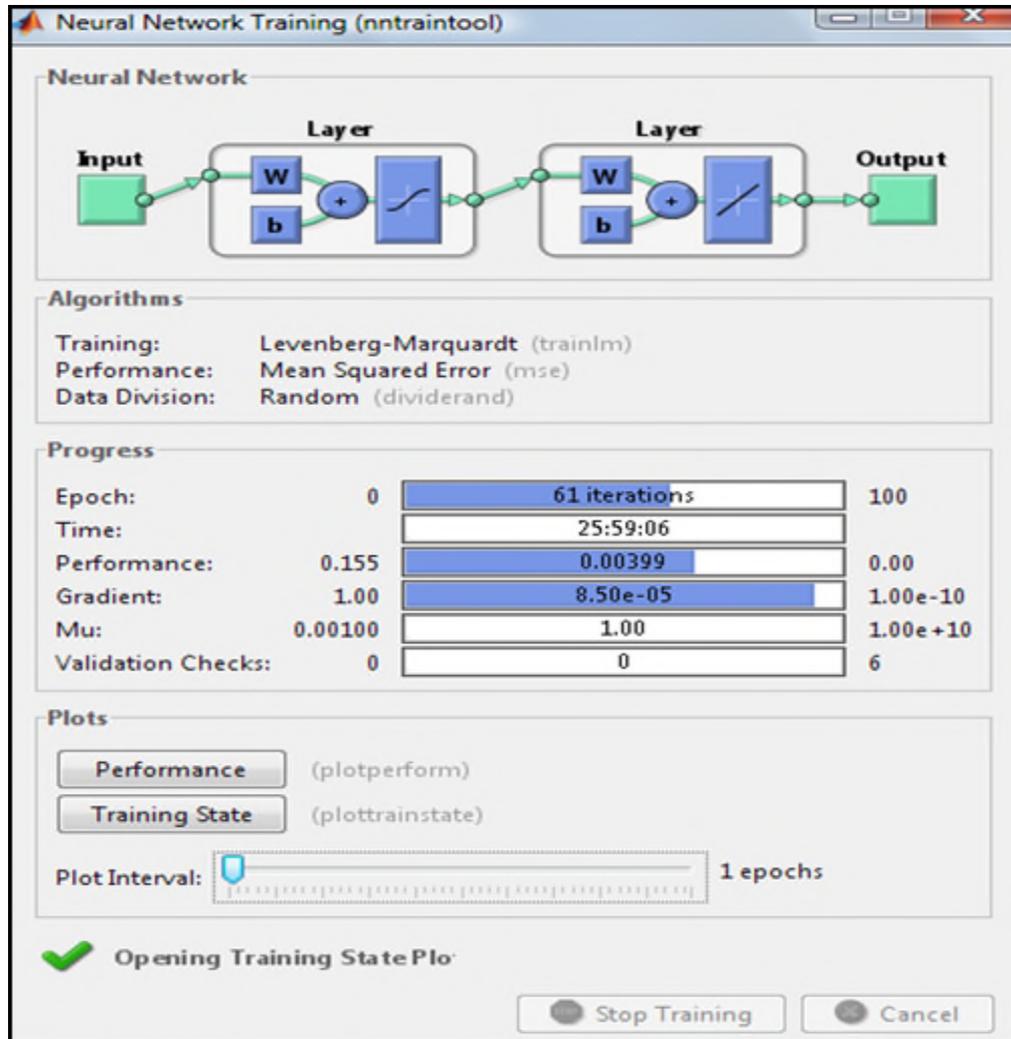


Figure IV-16: Aperçu du modèle

L'apprentissage est stoppé lorsque l'erreur (EQM) devient minimum, ou bien le nombre d'itération fixé atteint le maximum. Ensuite le réseau est évalué à partir des données différentes de celles utilisées durant l'apprentissage. Par définition ce dernier jeu de données est appelé ensemble test.

Le pas du gradient ainsi que le pas d'apprentissage sont des facteurs déterminant dans la vitesse de convergence du réseau de neurones. Le temps d'apprentissage croissant très vite avec la complexité du réseau, il faut donc trouver un pas optimal. Plus le pas est petit, plus le nombre d'itérations de la base d'apprentissage sera important. Tandis que plus le pas est grand, plus le nombre d'itérations nécessaire sera faible mais le réseau risque de

diverger. Dans le cadre de notre expérience, nous avons fixé le pas d'apprentissage à 0,01 et le nombre maximum d'itérations à 100. Les résultats sont mentionnés dans les expériences suivantes:

IV.6 EXPERIMENTATION ET RESULTATS:

IV.6.1 EXPERIENCE 1:(CARACTERES SIMILAIRES)

Dans cette expérience, on va comparer les lettres similaires c'est-à-dire les caractères aussi différents par la position et le nombre de point au-dessus ou en-dessous du trajectoire du caractère. Bien sur leur prononciation phonétique dissemblable par exemple (jim, ha, kha). Les caractères différents ont été retirés par exemple (Alef, Lam), Nous avons obtenus les partition en neuf groupes suivantes:

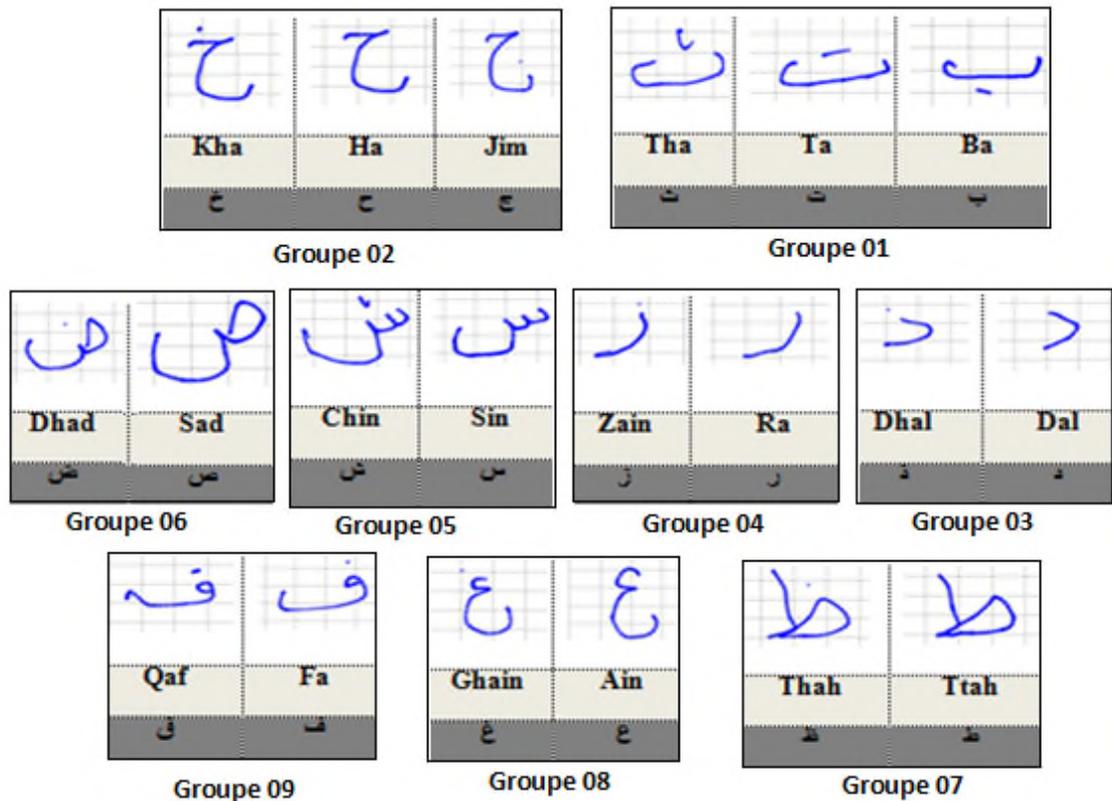


Figure IV-17: Découpage des neuf groupes d'apprentissage et illustration de la ressemblance entre caractères.

On rappelle que l'architecture idéale est la suivante:

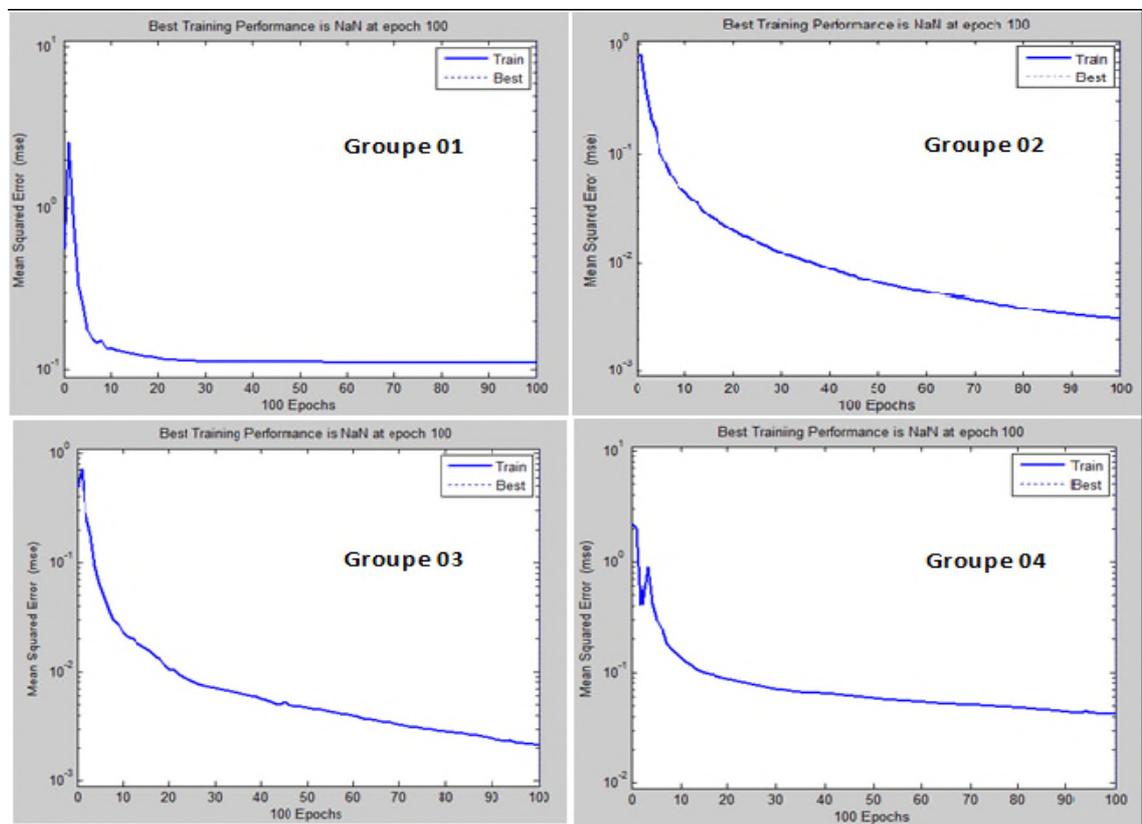
TDNN à 1 couche cachée pour la partie extraction et classifieur PMC linéaire:
 La fonction de d'apprentissage est :Rprop
Couche d'entrée du TDNN à poids partagés, l = 0:
 Nombre de neurones selon la direction temporelle, $window_T[0]= 17$
 Nombre de neurones selon la direction caractéristique, $nb_feat[0]= 7$
 Taille du champ perceptuel (fenêtre) par la couche, $field_T[0]= 5$
 Délai temporel (nombre de neurones) entre chaque fenêtre, $delay[0]= 3$

Couche cachée du TDNN, l=1 (couche d'entrée du classifieur PMC, $nn_l=0$)
 Nombre de neurones selon la direction temporelle, $window_T[1]= 8$
 Nombre de neurones selon la direction caractéristique, $nb_feat[1]= 14$

Couche de sortie PMC, $nn_l=2$
 Nombre de neurones $NN_nb_neuron[2]=$ nombre de classes (28)

Figure IV-18: Architecture idéale du réseau TDNN.

La figure suivante illustre les courbes représentant respectivement, l'Erreur Quadratique Moyenne ou en anglais Mean Squart Error:



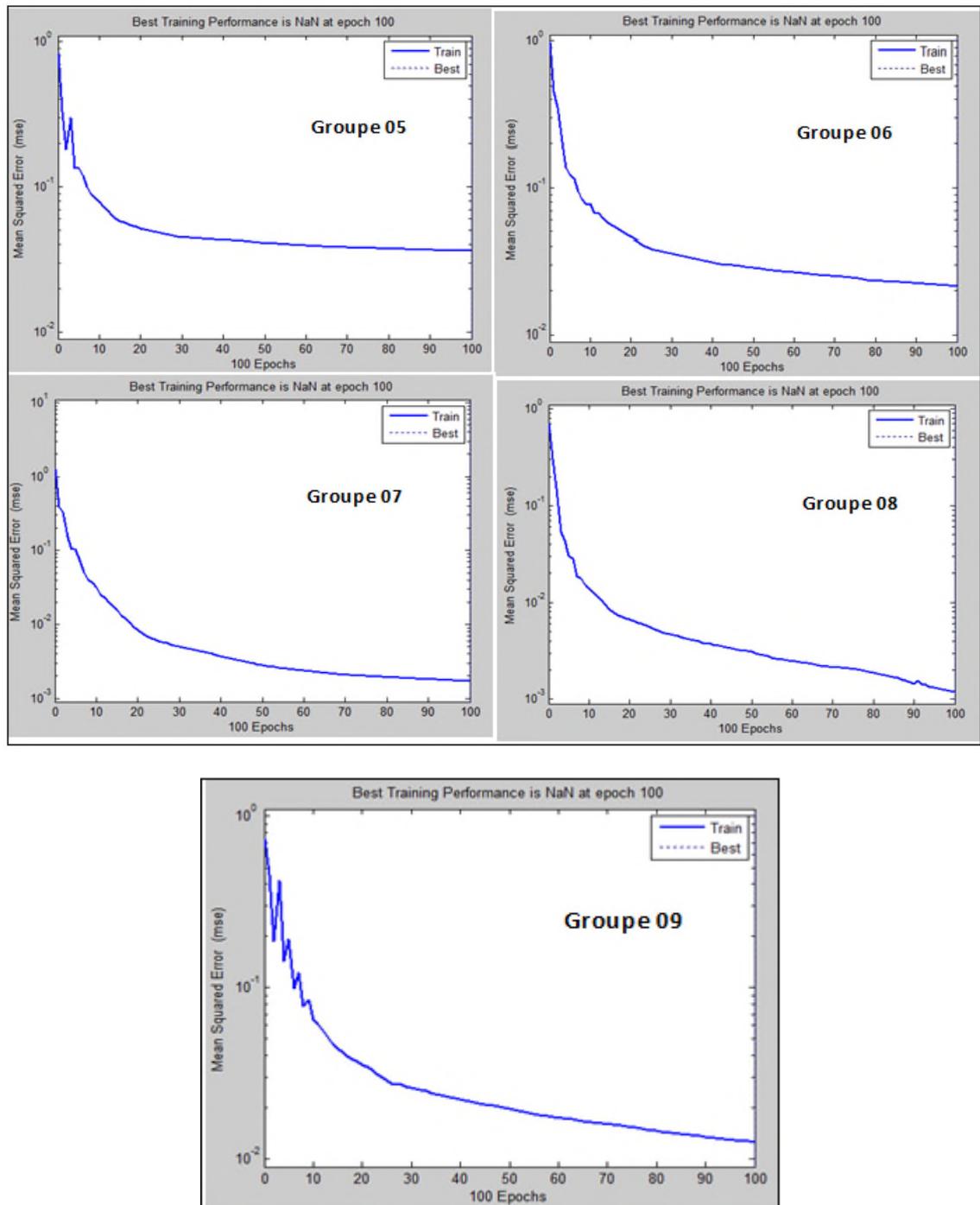


Figure IV-19: Sortie du réseau, EQM pour les 9 groupes

Autres résultats prenant en charge l'EQM, le taux d'apprentissage symbolisé par "C" et "Time" le temps d'exécution écoulé, ont été résumés dans le tableau suivant :

Tableau IV-1: Les taux de C, EQM et le temps d'exécution pour chaque groupe

	Le groupe1	Le groupe2	Le groupe3	Le groupe4	Le groupe5
EQM	11.10^{-2}	3.10^{-3}	2.10^{-3}	42.10^{-3}	36.10^{-3}
C	66.5 %	99.8 %	99.9 %	97.1 %	98.4 %
Time	5m16s	5m33s	3m25s	3m18s	3m25s
	Le groupe 6	Le groupe 7	Le groupe 8	Le groupe 9	
EQM	21.10^{-3}	1.10^{-3}	1.10^{-3}	12.10^{-3}	
C	98.8 %	99.9 %	99.9 %	99.6%	
Time	3m21s	3m16s	3m18s	3m16s	

La matrice de confusion (CM) regroupant le nombre d'échantillons bien classés dans la diagonale ainsi que la matrice de performance (Per) qui regroupe les échantillons mal-classés et ceux bien classés de chaque caractère sont illustrés pour les neuf groupe dans la figure IV-20.

<table border="1"> <tr><td>cm</td><td></td><td></td><td></td></tr> <tr><td>218</td><td>292</td><td>0</td><td></td></tr> <tr><td>218</td><td>292</td><td>0</td><td></td></tr> <tr><td>0</td><td>1</td><td>509</td><td></td></tr> </table>	cm				218	292	0		218	292	0		0	1	509		<table border="1"> <tr><td>per</td><td></td><td></td><td></td></tr> <tr><td>0,572</td><td>0,427</td><td>0,427</td><td></td></tr> <tr><td>0,427</td><td>0,574</td><td>0,572</td><td></td></tr> <tr><td>0,0019</td><td>0</td><td>0,998</td><td></td></tr> </table>	per				0,572	0,427	0,427		0,427	0,574	0,572		0,0019	0	0,998		Groupe 1	<table border="1"> <tr><td>cm =</td><td></td><td></td><td></td></tr> <tr><td>499</td><td>11</td><td></td><td></td></tr> <tr><td>9</td><td>501</td><td></td><td></td></tr> </table>	cm =				499	11			9	501			Groupe 5	<table border="1"> <tr><td>per =</td><td></td><td></td><td></td></tr> <tr><td>0.0216</td><td>0.0176</td><td>0.9784</td><td></td></tr> <tr><td>0.0176</td><td>0.0216</td><td>0.9824</td><td></td></tr> </table>	per =				0.0216	0.0176	0.9784		0.0176	0.0216	0.9824	
cm																																																													
218	292	0																																																											
218	292	0																																																											
0	1	509																																																											
per																																																													
0,572	0,427	0,427																																																											
0,427	0,574	0,572																																																											
0,0019	0	0,998																																																											
cm =																																																													
499	11																																																												
9	501																																																												
per =																																																													
0.0216	0.0176	0.9784																																																											
0.0176	0.0216	0.9824																																																											
<table border="1"> <tr><td>cm</td><td></td><td></td><td></td></tr> <tr><td>510</td><td>0</td><td>0</td><td></td></tr> <tr><td>1</td><td>509</td><td>0</td><td></td></tr> <tr><td>0</td><td>2</td><td>508</td><td></td></tr> </table>	cm				510	0	0		1	509	0		0	2	508		<table border="1"> <tr><td>per =</td><td></td><td></td><td></td></tr> <tr><td>0</td><td>0.0020</td><td>1.0000</td><td></td></tr> <tr><td>0.0020</td><td>0.0039</td><td>0.9980</td><td></td></tr> <tr><td>0.0039</td><td>0</td><td>0.9961</td><td></td></tr> </table>	per =				0	0.0020	1.0000		0.0020	0.0039	0.9980		0.0039	0	0.9961		Groupe 2	<table border="1"> <tr><td>cm =</td><td></td><td></td><td></td></tr> <tr><td>498</td><td>12</td><td></td><td></td></tr> <tr><td>0</td><td>510</td><td></td><td></td></tr> </table>	cm =				498	12			0	510			Groupe 6	<table border="1"> <tr><td>per =</td><td></td><td></td><td></td></tr> <tr><td>0.0235</td><td>0</td><td>0.9765</td><td></td></tr> <tr><td>0</td><td>0.0235</td><td>1.0000</td><td></td></tr> </table>	per =				0.0235	0	0.9765		0	0.0235	1.0000	
cm																																																													
510	0	0																																																											
1	509	0																																																											
0	2	508																																																											
per =																																																													
0	0.0020	1.0000																																																											
0.0020	0.0039	0.9980																																																											
0.0039	0	0.9961																																																											
cm =																																																													
498	12																																																												
0	510																																																												
per =																																																													
0.0235	0	0.9765																																																											
0	0.0235	1.0000																																																											
<table border="1"> <tr><td>cm =</td><td></td><td></td><td></td></tr> <tr><td>510</td><td>0</td><td></td><td></td></tr> <tr><td>1</td><td>509</td><td></td><td></td></tr> </table>	cm =				510	0			1	509			<table border="1"> <tr><td>per =</td><td></td><td></td><td></td></tr> <tr><td>0</td><td>0.0020</td><td>1.0000</td><td></td></tr> <tr><td>0.0020</td><td>0</td><td>0.9980</td><td></td></tr> </table>	per =				0	0.0020	1.0000		0.0020	0	0.9980		Groupe 3	<table border="1"> <tr><td>cm =</td><td></td><td></td><td></td></tr> <tr><td>510</td><td>0</td><td></td><td></td></tr> <tr><td>1</td><td>509</td><td></td><td></td></tr> </table>	cm =				510	0			1	509			Groupe 7	<table border="1"> <tr><td>per =</td><td></td><td></td><td></td></tr> <tr><td>0</td><td>0.0020</td><td>1.0000</td><td></td></tr> <tr><td>0.0020</td><td>0</td><td>0.9980</td><td></td></tr> </table>	per =				0	0.0020	1.0000		0.0020	0	0.9980									
cm =																																																													
510	0																																																												
1	509																																																												
per =																																																													
0	0.0020	1.0000																																																											
0.0020	0	0.9980																																																											
cm =																																																													
510	0																																																												
1	509																																																												
per =																																																													
0	0.0020	1.0000																																																											
0.0020	0	0.9980																																																											
<table border="1"> <tr><td>cm =</td><td></td><td></td><td></td></tr> <tr><td>507</td><td>3</td><td></td><td></td></tr> <tr><td>26</td><td>484</td><td></td><td></td></tr> </table>	cm =				507	3			26	484			<table border="1"> <tr><td>per =</td><td></td><td></td><td></td></tr> <tr><td>0.0059</td><td>0.0510</td><td>0.9941</td><td></td></tr> <tr><td>0.0510</td><td>0.0059</td><td>0.9490</td><td></td></tr> </table>	per =				0.0059	0.0510	0.9941		0.0510	0.0059	0.9490		Groupe 4	<table border="1"> <tr><td>cm =</td><td></td><td></td><td></td></tr> <tr><td>510</td><td>0</td><td></td><td></td></tr> <tr><td>1</td><td>509</td><td></td><td></td></tr> </table>	cm =				510	0			1	509			Groupe 8	<table border="1"> <tr><td>per =</td><td></td><td></td><td></td></tr> <tr><td>0</td><td>0.0020</td><td>1.0000</td><td></td></tr> <tr><td>0.0020</td><td>0</td><td>0.9980</td><td></td></tr> </table>	per =				0	0.0020	1.0000		0.0020	0	0.9980									
cm =																																																													
507	3																																																												
26	484																																																												
per =																																																													
0.0059	0.0510	0.9941																																																											
0.0510	0.0059	0.9490																																																											
cm =																																																													
510	0																																																												
1	509																																																												
per =																																																													
0	0.0020	1.0000																																																											
0.0020	0	0.9980																																																											
<table border="1"> <tr><td>cm =</td><td></td><td></td><td></td></tr> <tr><td>507</td><td>3</td><td></td><td></td></tr> <tr><td>1</td><td>509</td><td></td><td></td></tr> </table>	cm =				507	3			1	509			<table border="1"> <tr><td>per =</td><td></td><td></td><td></td></tr> <tr><td>0.0059</td><td>0.0510</td><td>0.9941</td><td></td></tr> <tr><td>0.0510</td><td>0.0059</td><td>0.9490</td><td></td></tr> </table>	per =				0.0059	0.0510	0.9941		0.0510	0.0059	0.9490		Groupe 9	<table border="1"> <tr><td>cm =</td><td></td><td></td><td></td></tr> <tr><td>507</td><td>3</td><td></td><td></td></tr> <tr><td>1</td><td>509</td><td></td><td></td></tr> </table>	cm =				507	3			1	509			Groupe 9	<table border="1"> <tr><td>per =</td><td></td><td></td><td></td></tr> <tr><td>0.0059</td><td>0.0020</td><td>0.9941</td><td></td></tr> <tr><td>0.0020</td><td>0.0059</td><td>0.9980</td><td></td></tr> </table>	per =				0.0059	0.0020	0.9941		0.0020	0.0059	0.9980									
cm =																																																													
507	3																																																												
1	509																																																												
per =																																																													
0.0059	0.0510	0.9941																																																											
0.0510	0.0059	0.9490																																																											
cm =																																																													
507	3																																																												
1	509																																																												
per =																																																													
0.0059	0.0020	0.9941																																																											
0.0020	0.0059	0.9980																																																											

Figure IV-20: Les matrices de confusion et de performance pour chaque groupe

La figure IV-21 montre les taux de généralisation des caractères :

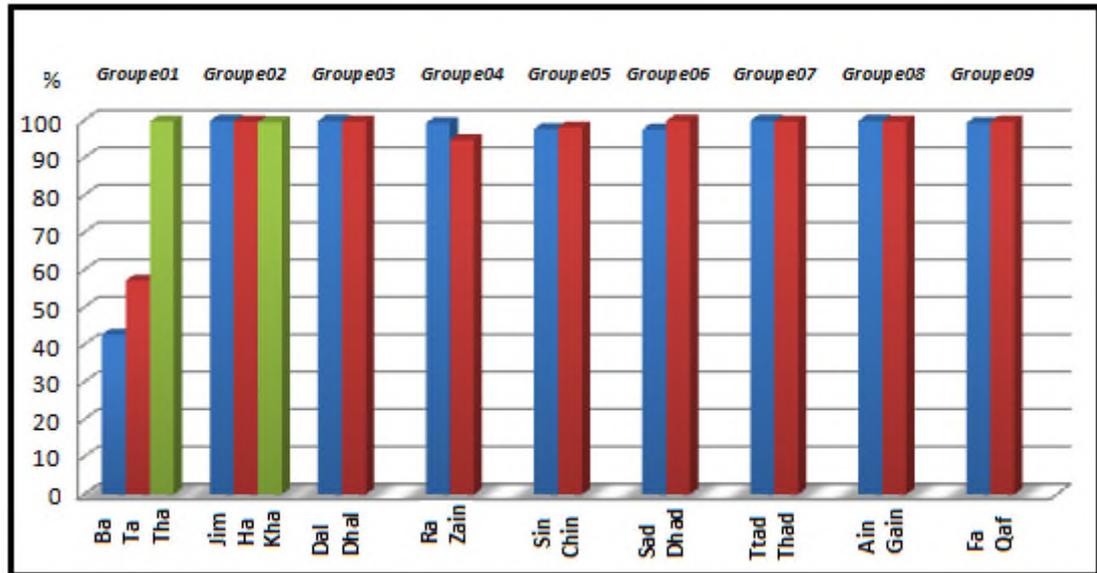


Figure IV-21: les taux de généralisation des caractères des neuf groupes

IV.6.2 DISCUSSION

Les résultats obtenus par la première expérience montrent une efficacité importante de discrimination pour la plupart des groupes en matière de confusion, par contre le premier groupe présente une insuffisance de classification se justifie par la position du point diacritique au-dessous du caractère "Ba", ce que n'est pas le cas pour les autres groupes où la position est resté inchangé, par exemple le neuvième groupe des caractère "Fa" et "Qaf".

Cette expérience nous a montré que le nombre de points n'influe pas sur la discrimination de ce modèle neuronal par contre la position des points montre parfois une influence remarquable à l'échelle expérimental.

IV.6.3 EXPERIENCE 2:(CARACTERES DIFFERENTS)

La même architecture précédente est reconduite, mais en changeant la fonctions d'apprentissage de la Rétropropagation du gradient "résistante" (RP) avec l'algorithme Levenberg-Marquart (LM)

Dans cette expérience, on a comparé les lettres différentes par exemple (Alef, Ba, Ain).les caractères similaires ont été retirés par exemple (jim, ha, kha), donc choisir à partir de chacun des neuf groupes, un caractère En plus des caractères qui n'ont pas semblable .Nous avons eu après la partition3 groupes

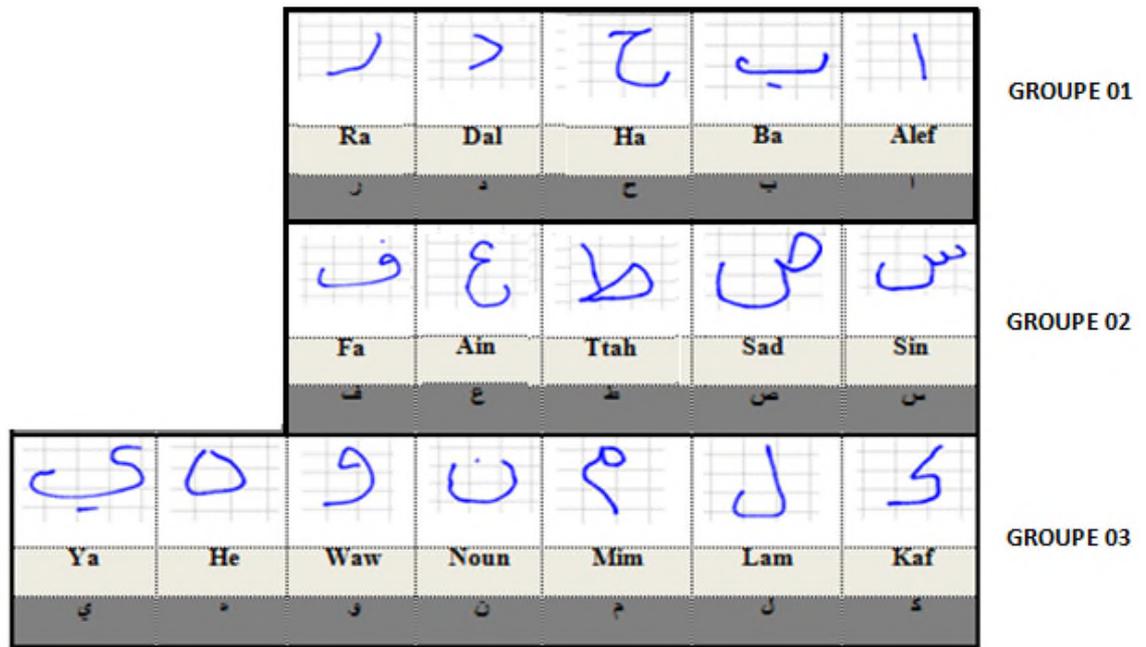


Figure IV-22: Découpage des trios groupes d'apprentissage et illustration de la différents entre caractères.

Tableau IV-2: Les taux de C, EQM et le temps d'exécution pour chaque groupe

	Le groupe 01	Le groupe 02	Le groupe 03
EQM	14.10 ⁻³	15.10 ⁻³	37.10 ⁻³
C	95.8 %	95.3%	80.67 %
Time	5m16s	5m33s	1h.09m08s

Voici les matrices de confusion et de performance des 4 groupes:

cm =					per =			Groupe 01		
510	0	0	0	0	0	0	1.0000			
0	510	0	0	0	0	0.0039	1.0000			
0	2	445	63	0	0.1275	0.0784	0.8725			
0	0	40	470	0	0.0784	0.1235	0.9216			
0	0	0	0	510	0	0	1.0000			
cm =					per =			Groupe 02		
506	4	0	0	0	0.0078	0.0039	0.9922			
2	507	0	0	1	0.0059	0.0078	0.9941			
0	0	454	56	0	0.1098	0.1098	0.8902			
0	0	56	454	0	0.1098	0.1098	0.8902			
0	0	0	0	510	0	0.0020	1.0000			
cm =					per =			Groupe 03		
502	0	2	4	0	0	2	0,02549		0,0039	0,9745
2	504	1	0	0	1	2	0,04706		0,9549	0,9529
0	2	448	60	0	0	0	0,13333		0,1	0,8667
0	0	46	464	0	0	0	0,02353		0,1255	0,9765
1	5	1	0	20	344	139	0,03922		0,0647	0,9608
1	0	0	0	4	473	32	0,99804		0	0,002
2	1	0	0	2	97	408	0,08627	0,1039	0,9137	

Figure IV-23: Les matrices de confusion et de performance pour chaque groupe

La figure IV-24 montre les taux de généralisation des caractères:

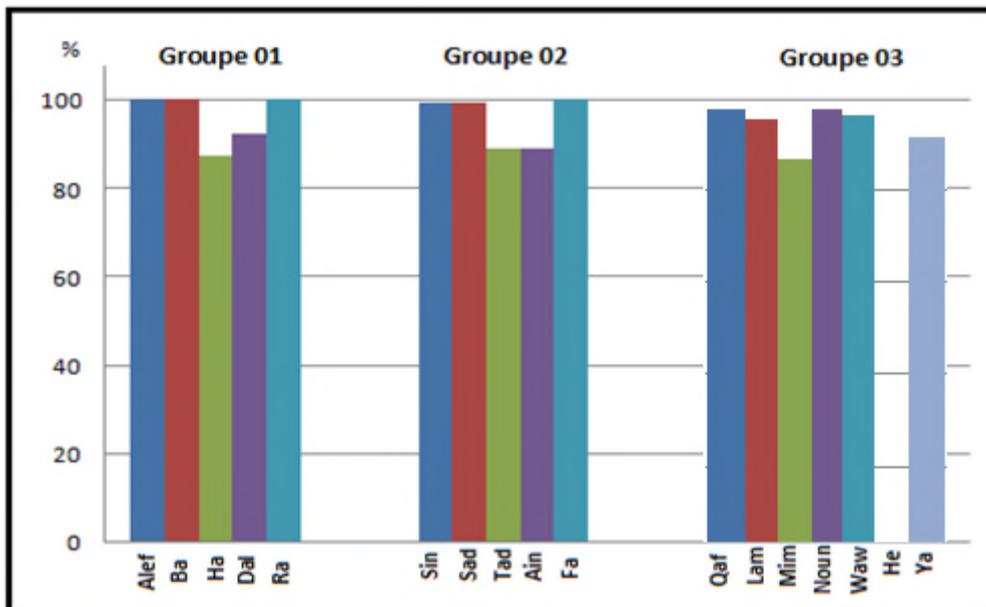


Figure IV-24: les taux de généralisation des caractères des trios groupes

IV.6.4 DISCUSSION

Les résultats obtenus par cette expérience montrent une stabilité de reconnaissance avec un taux assez important mais quelques perturbation de classification comme remarqué au niveau de caractère "He" Des meilleures résultats au niveaux des premiers caractère comme le "Alef", "Ba", le "Ra" et le "Sad".

Cela se justifie par l'apprentissage du TDNN basé sur l'adaptation des poids pour les derniers caractères du même groupe.

IV.6.5 EXPERIENCE 3:(LES 28 CARACTERES)

La même architecture précédente est reconduite, mais en variant les fonctions d'apprentissage

Dans cette section on a essayé de voir l'influence de la méthode de Levenberg-Marquart sous plusieurs versions sur le modèle.

Nous avons testé deux algorithmes du premier ordre : et la Rétropropagation du gradient "résistante" (RP) et l'algorithme Levenberg-Marquart (LM) deux algorithmes du second ordre, en l'occurrence : la Rétro-Propagation du Gradient Descendant (GD) et l'algorithme du gradient conjugué (SCG).

L'algorithme d'apprentissage requiert des besoins énormes de mémoire, la taille de la matrice d'entrée qui se propage dans l'architecture du TDNN, elle aussi ne facilitant pas la tâche. Ces raisons nous ont poussées à diviser l'apprentissage général en quatre parties.

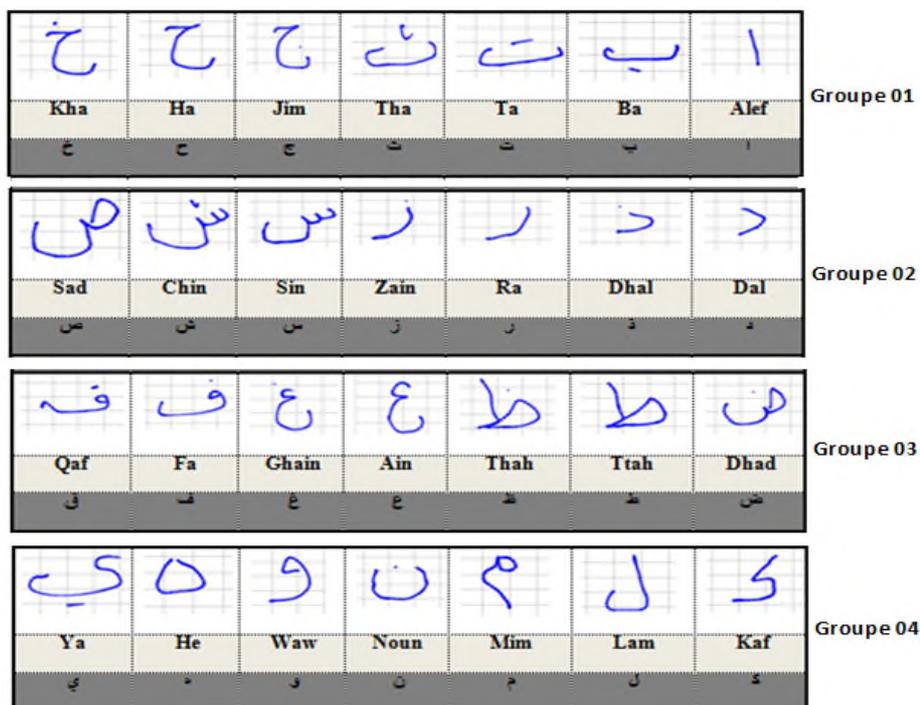


Figure IV-25 : Découpage des quatre groupes d'apprentissage

:Le tableau suivant récapitule les différents résultats obtenus:

Tableau IV-3: Résultats des variations de la fonction d'apprentissage

		RP	LM	GD	SCG
Groupe 01	EQM	61. 10 ⁻³	46. 10 ⁻³	600. 10 ⁻³	67. 10 ⁻³
	C	67.00%	72.07%	18.45%	61.23%
	Time	21m.07s	1h.07m.51s	21m.25s	43m.08s
Groupe 02	EQM	66. 10 ⁻³	29. 10 ⁻³	359. 10 ⁻³	91. 10 ⁻³
	C	67.11%	97.17%	18.03%	43.38%
	Time	21m.43s	1h.08m.35s	21m.25s	42m.54s
Groupe 03	EQM	66. 10 ⁻³	50. 10 ⁻³	418. 10 ⁻³	58. 10 ⁻³
	C	69.94%	70.00%	13.58%	69.97%
	Time	22m.12s	1h.07m.17s	31m.31s	44m.18s
Groupe 04	EQM	64. 10 ⁻³	37. 10 ⁻³	297. 10 ⁻³	54. 10 ⁻³
	C	69.91%	80.67%	72.07%	75.51%
	Time	21m.03s	h1.09m.08s	22m.13s	43m.46s

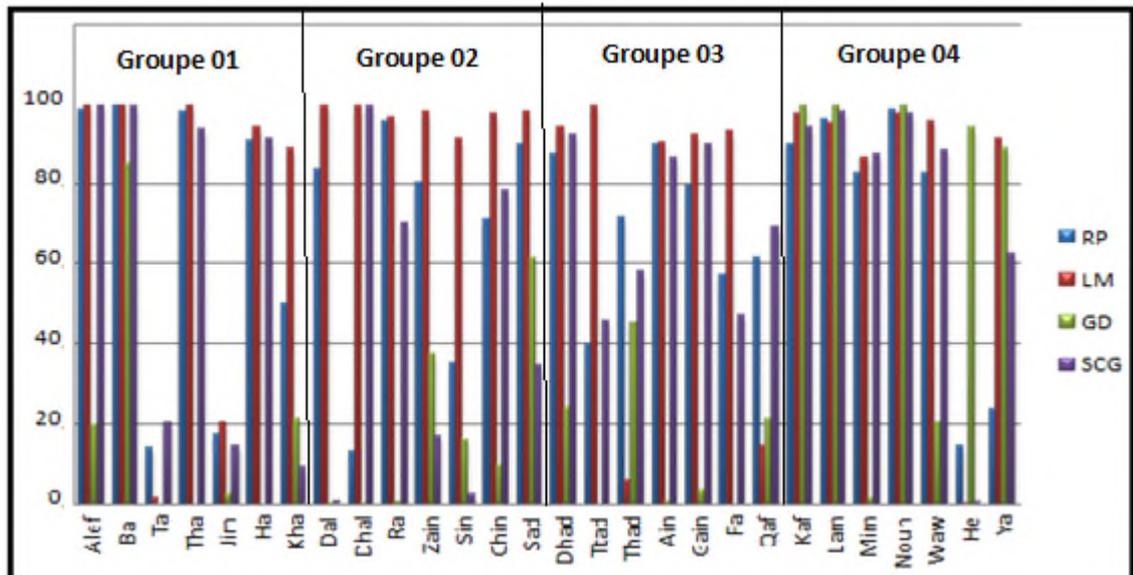


Figure IV-26: les taux de généralisation des 28 caractères avec quatre algorithmes d'apprentissage

VI.6.6 DISCUSSION:

L'algorithme "LM" a obtenu l'erreur quadratique moyenne la plus basse et un taux d'échantillon bien classés (C) très haut à tous les groupes, il est plus performant que les autres algorithmes, au détriment de son temps d'exécution plus ou moins plus élevé. Néanmoins, nous ne pouvons pas porter un jugement définitif sur cet algorithme qu'une fois sa capacité de généralisation confirmée.

On remarque que les taux de reconnaissance pour les groupes 1 et 3 sont relativement similaires, tandis que ceux des autres groupes 2 et 4 sont plus ou moins élevés, avec une mention spéciale pour le groupe 2 renfermant les lettres (Dal, Dhal, Ra, Zain, Sin, Chin, Sad) qui a atteint les 70%.

L'avantage de notre application fondée sur le modèle TDNN, est que celle-ci permet une exécution en ligne plus rapide.

VI.7 PHASE DE GENERALISATION:

La première étape à réaliser consisté en la construction de la base de données contenant des échantillons de lettres arabes. En effet, dans notre première version, une stratégie minimale a été effectuée pour réaliser le système de reconnaissance. Des boites de bondissement normalisées ont été préparées pour faciliter la tâche au scripteur lors de la saisie.

Après cette phase d'acquisition, tour à tour des opérations de prétraitement (échantillonnage spatial et recentrage du caractère) et d'extraction de caractéristique sont été effectuées. La matrice résultante de cette dernière opération va se propager dans le réseau TDNN.

Les expériences faites précédemment, nous ont permis de définir l'architecture idéale de notre réseau TDNN. Des contraintes de mémoire dues au matériel utilisé d'une part, et d'autre part a la divergence des algorithmes d'apprentissage dès que le nombre de classes (de caractères) augmente, ne nous ont pas permis de faire l'apprentissage des 28 caractères au même temps, d'où le découpage imposé des quatre groupes regroupant chacun sept caractères. Néanmoins, le découpage a été réalisé de tel façon à pouvoir comparer le caractère test aux autres classes voisines qui lui ressemblent, (exemple: Ra et Zin). Ces contraintes se sont répercutées sur la phase de généralisation et de test.

Pour une meilleure compréhension, nous avons conçus une interface graphique, permettant au système de regrouper les étapes; d'acquisition, d'extraction de la matrice de caractéristiques et surtout la classification, comme le montre la figure suivante:

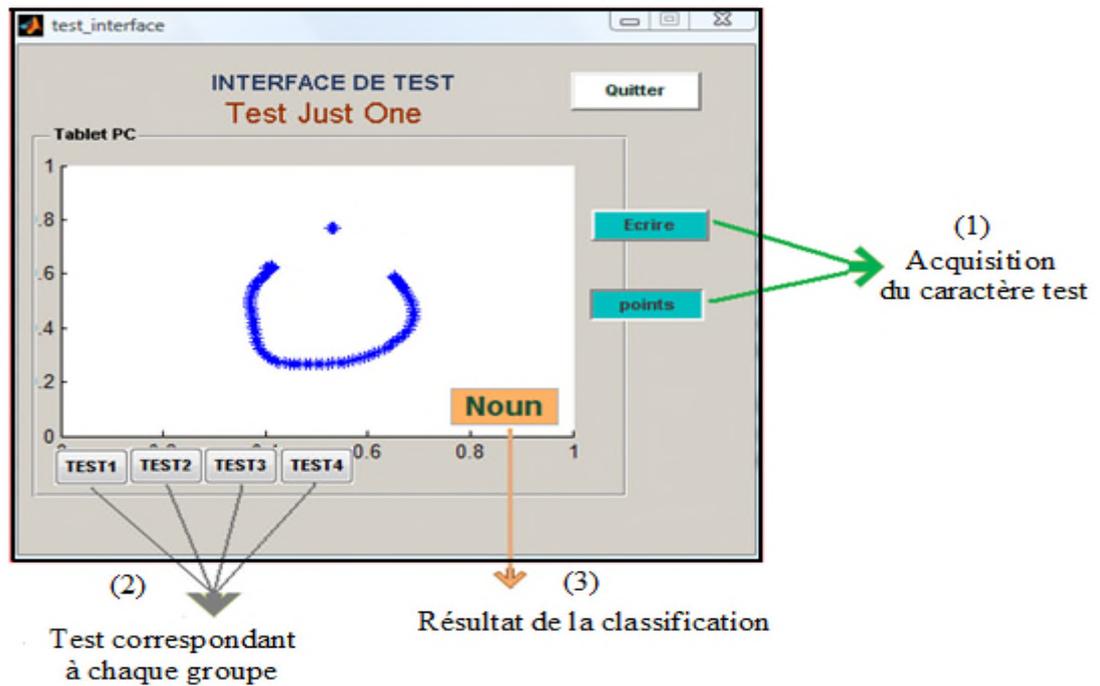


Figure IV-27: Test du caractère en ligne "Noun".

Rappelons que le but de ce travail, est de faire une acquisition, une extraction ainsi qu'une reconnaissance de caractère arabe d'une façon dynamique ou plutôt en ligne et cela au delà de la qualité du taux de reconnaissance des caractères discutable surtout quand il s'agit de caractères voisins qui se ressemblent.

CONCLUSION GENERALE ET PERSPECTIVES

L'objectif de ce travail est de contribuer à progresser dans le difficile problème de la reconnaissance de l'écriture manuscrite en-ligne, pour cela notre objectif consiste à développer une solution à base de réseau de neurone de type TDNN.

L'apport du but se fera à la fois sur l'étude, la conception et la mise en œuvre d'un type spécifique de réseaux de neurones aux propriétés très intéressantes, les réseaux à convolution(les TDNN's).

Ces réseaux sont spécifiquement destinés à traiter les formes 2D, et aptes à prendre leur variabilités en compte.

La reconnaissance des caractères manuscrits issus d'une tablette digitale et d'un stylet et souvent associé au domaine temporel.

Le type d'approche temporelle telle que le TDNN, présente l'intérêt d'avoir une représentation compacte et avantageuse d'un point de vue calcul, mais reste sensible à l'ordre des traits.

L'implémentation de cette méthode est bien détaillée par la suite dans ce mémoire, accompagnée des résultats obtenus.

Comme travail futur, le passage vers le deuxième stade en traitant les mots et les phrases représente un challenge intéressant surtout avec un enrichissement assuré de notre nouvelle base d'écriture arabe en ligne. Le but de ce mémoire est de contribuer dans l'essor terrible (minime sur le manuscrit arabe) qu'a connus le domaine de reconnaissance de l'écriture manuscrite non contrainte. Nous comptons aussi intégrer cette partie dans des thèmes plus élargis et plus complexes tels que la reconnaissance de mots en utilisant les contraintes contextuelles et lexicales.

Finalement, le but de la reconnaissance de l'écriture manuscrite est de pouvoir utiliser cette représentation dans une application informatique. De ce point de vue, les difficultés de la reconnaissance d'écriture sont liées bien évidemment à la variabilité des écritures, mais aussi au fait qu'on ne sait pas encore très bien modéliser l'énorme masse de connaissances et d'informations utiles à la reconnaissance. Notre système pourra mieux s'adapter en jouant sur d'autres paramètres du réseau qui nous offre la motivation d'attaquer d'autres défis et problématiques de ce domaine.

BIBIOPHAPHIE

- [1] H. Mouchère, "Étude des mécanismes d'adaptation et de rejet pour l'optimisation de classifieurs: Application à la reconnaissance de l'écriture manuscrite en-ligne," in *thèse de l'Institut National des Sciences Appliquées de Rennes*, 2007.
- [2] S. Quiniou, "Intégration de connaissances linguistiques pour la reconnaissance de textes manuscrits en-ligne," in *thèse de L'INSA DE RENNES*, 2007.
- [3] É. Poisson, "Architecture et apprentissage d'un système hybride neuro-markovien pour la reconnaissance de l'écriture manuscrite en-ligne," in *thèse de l'Université de Nantes*, 2005.
- [4] V. Margner and Haikal El Abed. "Arabichandwriting recognition competition". In ICDAR, pages 1274–1278. IEEE Computer Society, 2007.
- [5] M. Pechwitz, S. SnoussiMaddouri, V. Maergner, N. Ellouze, and H. Amiri. "Ifn/enit-database of handwrittenarabicwords". in the 7th Colloque International Francophone sur l'Écrit et le Document , CIFED 2002, Oct. 21-23, 2002, Hammamet, Tunis, (2002).
- [6] F. Slimane, S. Kanoun, J. Hennebert, A. M. Alimi et R. Ingold, "Modèles de Markov Cachés et Modèle de Longueur pour la reconnaissance de l'Écriture Arabe à Basse Résolution". MajecSTIC 2009. Avignon, France. 16-18 nov.2009
- [7] N. Ben Amara, A. Belaïd, N. Ellouze, "Modélisation pseudo bidimensionnelle pour la reconnaissance de chaînes de caractères arabes imprimé." Proc. 1er Colloque international francophone sur l'écrit et le document (CIFED'98), pp. 131-140, Québec, Canada, 1998.
- [8] H. Miled : "Stratégies de reconnaissance de l'écriture semi cursive : application aux mots manuscrits arabes". Thèse de doctorat, Université de Rouen, 1998.
- [9] N. Ben Amara, A. Belaïd et N. Ellouze, "Utilisation des modèles markoviens en reconnaissance de l'écriture arabe : État de l'art." Colloque International Francophone sur l'Écrit et le Document - CIFEd'00 (2000)
- [10] S. Al-OsaimiE.Atwell, L. Al-Sulaiti and B. Abu Shawar. "A review of arabic corpus analysis tools". In Actes de TALN'04, Fès, Maroc, 2004.
- [11] F. Menasri, "Contributions à la reconnaissance de l'écriture arabe manuscrite," in *thèse de l'Université Paris Descartes*, 2008.
- [12] N. Kharma, M. Ahmed, and R. Ward. "A new comprehensive database of handwritten arabic words, numbers, and signatures used for ocr testing". Electrical and Computer Engineering, 1999 IEEE Canadian Conference on, 2 :pp 766–768 vol.2, 1999.

- [13] S. Al-Maàdeed, D. Elliman, and C. A. Higgins. "A data base for arabic handwritten text recognition research". *The International Arab Journal of Information Technology*, Vol. 1, No. 1, January 2004.
- [14] Y. Al-Ohali, M. Cheriet, and C. Y. Suen. "Databases for recognition of handwritten arabic cheques". *Pattern Recognition*, vol 36(1) :pp 111–121, 2003
- [15] S. N. Srihari, G. R. Ball and H. Srinivasan, "Versatile Search of Scanned Arabic Handwriting," *Arabic and Chinese Handwriting Recognition*, LNCS 4768, Springer 2008, pp. 57-69
- [16] A. Amin. "Offline arabic character recognition" - a survey. In ICDAR '97 :Proceedings of the 4th International Conference on Document Analysis and Recognition, pp 596–599. IEEE Computer Society, 1997.
- [17] N. Ben Amara, A. Belaid, and N. Ellouze. "Utilisation des modèles markoviens en reconnaissance de l'écriture arabe état de l'art". In Colloque International Francophone sur l'Écrit et le Document (CIFED'00), Lyon, France, juillet 2000.
- [18] L. M. Lorigo and V. Govindaraju. "Offline arabic handwriting recognition" : A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 28(5) :pp 712–724, 2006.
- [19] A. Belaid and C. Choisy. "Human reading based strategies for off-line arabic word recognition. Summit on Arabic and Chinese Handwriting Recognition" 2006 - SACH'06, 2006.
- [20] J. s L. McClelland and D. E. Rumelhart. "An interactive activation model of context effects in letter perception" : Part 1. An account of basic findings. *Psychological Review*, 88(5) :pp 375–405, 1981.
- [21] M. Cheriet. "Strategies for visual arabic handwriting recognition : issues and case study". In ISSPA 2007, International Symposium on Signal Processing and its Applications, 12 - 15 February 2007, Sharjah, United Arab Emirates, Feb 2007.
- [22] K. Hallouli, "Reconnaissance de caractères par méthodes markoviennes et réseaux bayésiens", Thèse de l'École nationale supérieure des telecommunications, Paris mai 2004.
- [23] S. R. Ramesh, "A generalized character recognition algorithm: A graphical approach". *Pattern recognition*, 22, pp 347-350.
- [24] F. Lebourgeois, "Approche mixte pour la reconnaissance des documents imprimés", Institut National des Sciences Appliquées de Lyon, thèse de 3e cycle 1991.
- [25] J. Pearl. "Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference" Second edition, Morgan Kaufmann, San Mateo, Calif, 1988.
- [26] S. Souafi, "Contribution a la reconnaissance des structures des documents écrits; approche probabiliste", INSA de Lyon, 2002.

- [27] X. Xiao et G. Leedham, "Signature verification using a modified bayesian network". *Pattern recognition*, 35, pp 983-995.
- [28] K. Daoudi, D. Fohr, C. Antoine. "Réseaux Bayésiens pour la Reconnaissance Multi-bandes de la Parole" in XXIVème Journées d'Études sur la Parole, Nancy Juin 2002.
- [29] H. El Abed, V. Märgner, M. Kherallah, et A. M. Alimi, "Icdar 2009 online Arabichandwriting recognition competition," in *2009 10th International Conference on Document Analysis and Recognition*, p. 1388–1392, 2009
- [30] V. Vapnik, "The Nature of Statistical Learning Theory", Springer-Verlag, 1995.
- [31] W.S McCulloch, W.A.Pitts "A logical calculus of the ideas immanent in nervous activity", *Bull. Math. Biophys.* Volume 5. 1943.
- [32] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K. Lang : "Phonème recognition using time delay neural networks". 1987
- [33] E. H. Ratzlaff, "Methods, report and survey for the comparison of diverse isolated character recognition results on the unipen database," *Document Analysis and Recognition journal*, vol. 1, n°. 2, p. 623, 2003.
- [34] M. Parizeau, A. Lemieux, et C. Gagné, "Character recognition experiments using UNIPEN data," in *Proceedings. Sixth International Conference on Document Analysis and Recognition, Seattle*, p. 481–485, 2001.
- [35] R. Tlemsaniet A. Benyettou, "On Line Isolated Characters Recognition Using Dynamic Bayesian Networks," *International Arab Journal of Information Technology*, vol. 08, n°. 04, Oct. 2011.
- [36] M. Riedmiller and H. Braun. "A direct adaptative method for faster backpropagation learning : The rprop algorithm". In *Proceedings of the IEEE International Conference on Neural Networks (ICNN93)*, San Francisco, April 1993.
- [37] C. Igel and M. Hüsken. "Empirical evaluation of the improved Rprop learning algorithm". *Neurocomputing*, 50(C) :105{123, 2003.
- [38] R. Tlemsani "Reconnaissance En-Ligne du Manuscript Arabe" Thèse de l'université des sciences et de la technologie des sciences-MOHAMED BOUDIAF- d'Oran. 2012