

Réseaux de neurones à convolution

Reconnaissance de l'écriture manuscrite non contrainte

Emilie POISSON, Christian VIARD-GAUDIN

Ecole polytechnique de l'université de Nantes, France
IRCCyN UMR CNRS 6597 - Equipe Image et Vidéo Communications
e-mail : {emilie.poisson,christian.viardgaudin}@polytech.univ-nantes.fr

Résumé - *Ce papier présente l'étude, l'implémentation et le test d'un réseau de neurones particulier le TDNN - Time Delay Neural Network – réseau de neurones à décalage temporel, dans le cadre de la reconnaissance de l'écriture manuscrite. Les problèmes abordés concernent le choix de la topologie, du contrôle de la convergence et des algorithmes associés. Nous présentons les résultats obtenus sur deux bases de caractères isolés (IRONOFF et UNIPEN) dans ce premier exposé de nos travaux de recherche, ceux-ci donnent déjà des résultats très prometteurs. L'objectif est ensuite d'intégrer ce reconnaisseur dans un schéma hybride pour mener à bien la reconnaissance au niveau mot puis au niveau phrase.*

Mots-Clés - Reconnaissance de caractères manuscrits isolés, reconnaissance en-ligne, réseau de neurones à convolution, TDNN, SDNN.

1. Introduction

Les progrès réalisés ces dernières années dans le domaine de la reconnaissance de l'écriture manuscrite sont dus en grande partie à l'utilisation d'approches statistiques. Parmi celles-ci, deux techniques ont plus particulièrement été mises à contribution. Il s'agit d'une part des approches connexionnistes et d'autre part des modélisations basées sur les modèles de Markov cachés [20]. Les premières sont particulièrement intéressantes par leur fort pouvoir discriminant et leur capacité à construire des frontières de décisions complexes dans des espaces de grande dimension, les secondes modélisent par une approche paramétrique les séquences d'observations générées par des processus stochastiques tels que, par exemple, l'écriture manuscrite. La notion de séquence prend toute son importance lorsque l'on s'intéresse à la reconnaissance au niveau mot. Les HMMs -Hidden Markov Models, Modèles de Markov cachés - apportent alors leur capacité à modéliser la distribution des observations pour chaque classe de forme à reconnaître. Alors que, lorsqu'il s'agit de la reconnaissance de caractères isolés, c'est d'avantage une forme globale qu'il s'agit d'apprécier, les réseaux de neurones sont alors particulièrement adaptés. Certaines approches, parmi les plus évoluées, tentent de combiner les avantages de chacune de ces techniques en proposant des systèmes de reconnaissance hybrides [3, 7, 8, 17] : le réseau de neurones calcule les probabilités $P(\text{classe}/\text{observation})$, les séquences d'observation résultant de la concaténation de ces probabilités sont traitées par les HMMs. L'inconvénient de ce type d'approches est que la nature picturale 2D de l'écriture est mal prise en compte, les HMMs étant des structures de chaînes monodimensionnelles. Pour pallier à cette limitation des propositions sont apparues, elles consistent en une extension à des HMMs pseudo-2D [1, 9], voire à des champs de Markov, ces approches induisent des charges de calcul extrêmement lourdes et ne sont pas encore compatibles avec des approches temps réel.

Dans ce cadre, nous proposons d'étudier le comportement d'un type particulier de réseau qui possède d'intéressantes propriétés de robustesse vis-à-vis en particulier de la translation, il s'agit des TDNNs (Time Delay Neural Networks). Ils ont déjà montré leurs très grandes performances dans la reconnaissance de lettres isolées [10, 14], ils doivent pouvoir contribuer aussi à la reconnaissance mot [4, 22], voire phrase [18].

C'est l'objet des travaux qui vont être développés dans le cadre de cette thèse («Réseaux à convolution et approches statistiques hiérarchiques pour la reconnaissance de l'écriture»), thèse en phase de démarrage). De nombreux travaux ont déjà été menés sur les réseaux de neurones à convolution appliqués à la reconnaissance de l'écriture manuscrite [6, 22] mais restent en général limités à la reconnaissance de caractères ou chiffres pour des applications de type prise de notes. Ici, il s'agira d'étendre ces travaux à une reconnaissance plus globale de type mot voire phrase pour des applications orientés vers l'analyse de pages saisies en-ligne.

Nous avons dans un premier temps implémenté une architecture complète de TDNN que nous avons testée en limitant la reconnaissance à des caractères isolés afin de bien maîtriser les différents concepts de base des réseaux à convolution et de la reconnaissance d'écriture manuscrite en-ligne. Les problèmes abordés concernent le choix de la topologie, du contrôle de la convergence et des algorithmes associés. Nous présentons les résultats obtenus dans ce premier exposé de nos travaux de recherche.

2. Caractérisation d'un système de reconnaissance automatique d'écriture manuscrite

Rappelons que dans le domaine de la reconnaissance de l'écriture manuscrite deux domaines distincts sont considérés. Il s'agit d'une part de la reconnaissance statique, dite encore « hors-ligne », où les données disponibles résultent de la numérisation d'un document papier contenant de l'écriture. On dispose alors d'une image numérique où il va falloir avant même d'aborder l'étape de reconnaissance, localiser et extraire le signal d'écriture du reste de l'image. L'autre domaine est celui de la reconnaissance dynamique, appelée encore « en-ligne ». Cette fois, le signal d'écriture est directement enregistré pendant sa production grâce à un dispositif spécifique (assistant personnel, tablette graphique) qui permet d'échantillonner la trajectoire du stylet fournissant ainsi les coordonnées $x(n\Delta)$ et $y(n\Delta)$. Nous nous intéressons ici à la reconnaissance dans le domaine en-ligne, celle-ci connaît un regain d'intérêt manifeste avec l'avènement de la société de l'information dans laquelle nous entrons avec en particulier le besoin de mobilité et d'accès à l'information sans discontinuer.

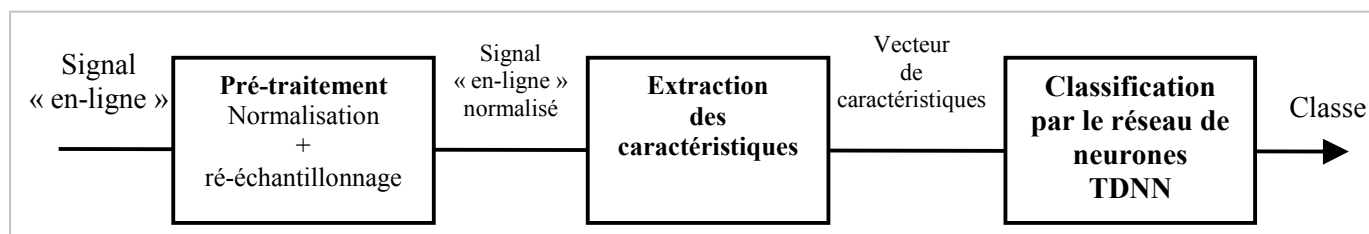


Figure 1 : Synoptique du système de reconnaissance de caractères.

Le système de reconnaissance proposé peut être représenté schématiquement selon la figure 1. Il comporte un étage de pré-traitement permettant de normaliser le caractère en taille et de ré-échantillonner son tracé en un nombre fixé de points équidistants. La seconde étape consiste en l'extraction de caractéristiques à partir du tracé précédemment obtenu, cette description constitue l'entrée du réseau de neurones. Celui-ci va fournir sur sa couche de sortie, une fois l'apprentissage réalisé, la classe du caractère présenté à l'entrée.

3. Les réseaux à convolution

Les réseaux de neurones de type TDNN sont des réseaux de type convolutif de par leur topologie, ils comportent une fenêtre glissante correspondant à un champ de vision restreint du signal global. Ils ont été utilisés initialement en reconnaissance de la parole [24], mais ils ont depuis également été utilisés avec succès dans la reconnaissance de caractères isolés tels que les chiffres ou même les mots [4].

La figure 2 illustre les principales différences entre un réseau de neurones classique de type perceptron multi-couches [5] et un réseau de neurones à convolution.

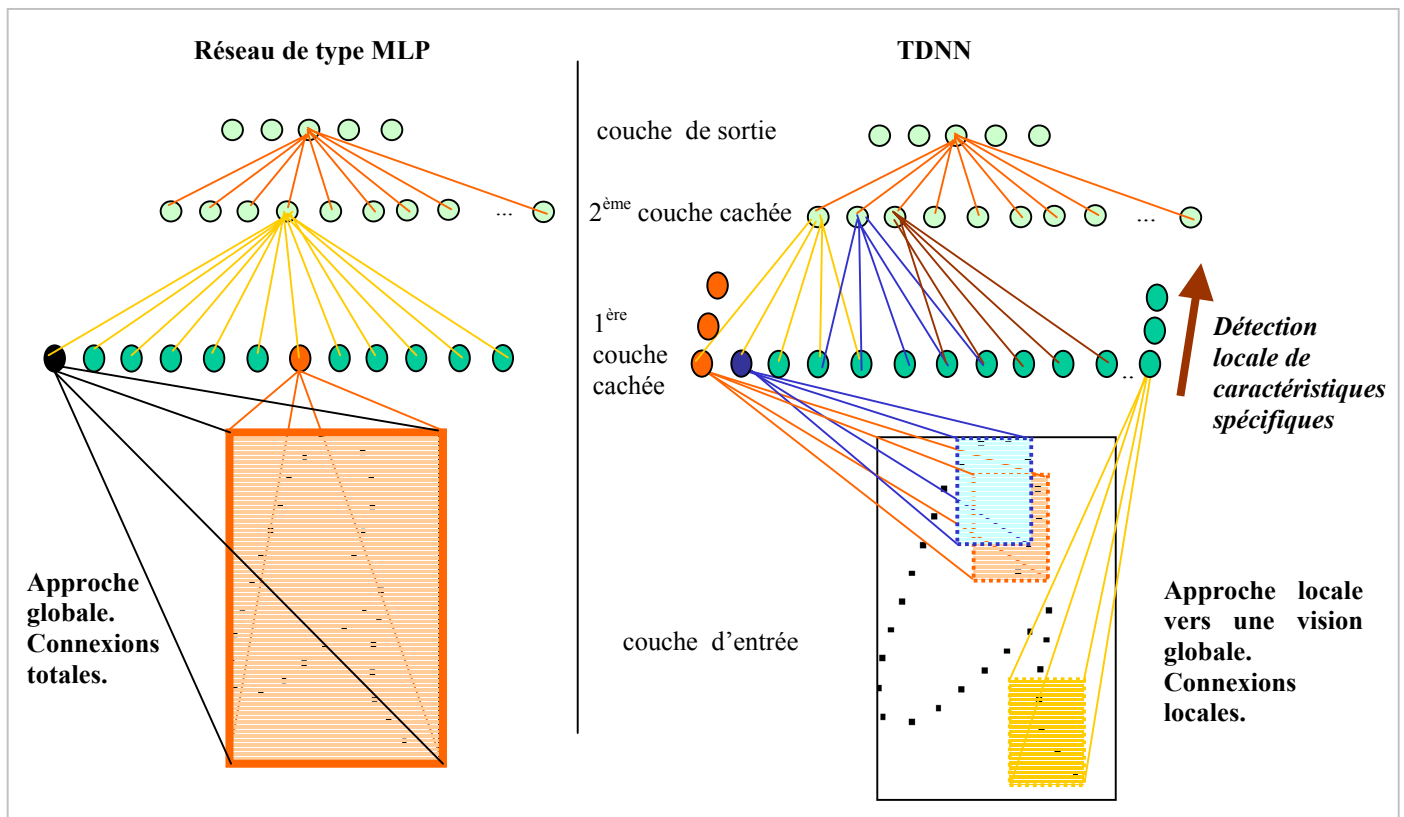


Figure 2 : Illustration de la différence Perceptron/TDNN sur le caractère 'a'.

Un neurone d'une couche d'un perceptron est connecté à tous les neurones de la couche précédente tandis que pour un réseau de neurones à convolution un neurone est connecté à un sous-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 à convolution incorporent des contraintes et réalisent un certain degré d'invariance de décalage et de déformation en utilisant trois idées : **zones réceptives locales, poids partagés, et sous-prélèvement spatial**. L'utilisation des poids partagés réduit le nombre de paramètres dans le système facilitant la généralisation. Ce type de réseau a été avec succès appliqué à la reconnaissance des chiffres [10].

Le principe de convolution impose une contrainte de partage des poids. Le comportement présumé du cerveau humain utilise ce concept de poids partagés [2]. En effet 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. On a plusieurs neurones qui calculent la même fonction sur des entrées différentes.

Deux grands types de réseaux à convolution sont actuellement développés pour la reconnaissance de caractères manuscrits : les TDNNs, Time Delay Neural Networks (soit réseaux de neurones à délais temporels) et les SDNNs, Space Displacement Neural Networks (soit réseaux de neurones à déplacement spatial). Le TDNN est un réseau à délai utilisé pour des données de nature séquentielle, donc particulièrement adapté à la reconnaissance de l'écriture en-ligne tandis que les SDNNs [13, 14] sont dédiés à des données de nature spatiale.

4. Le TDNN

4.1. Architecture et modélisation d'un TDNN.

L'architecture retenue comporte deux parties principales. La première, correspondant aux couches basses, implémente les convolutions successives permettant de transformer progressivement les caractéristiques en grandeurs de plus en plus significatives vis-à-vis du problème. La seconde correspond à un MLP (Multi Layer Perceptron) classique [5], 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.

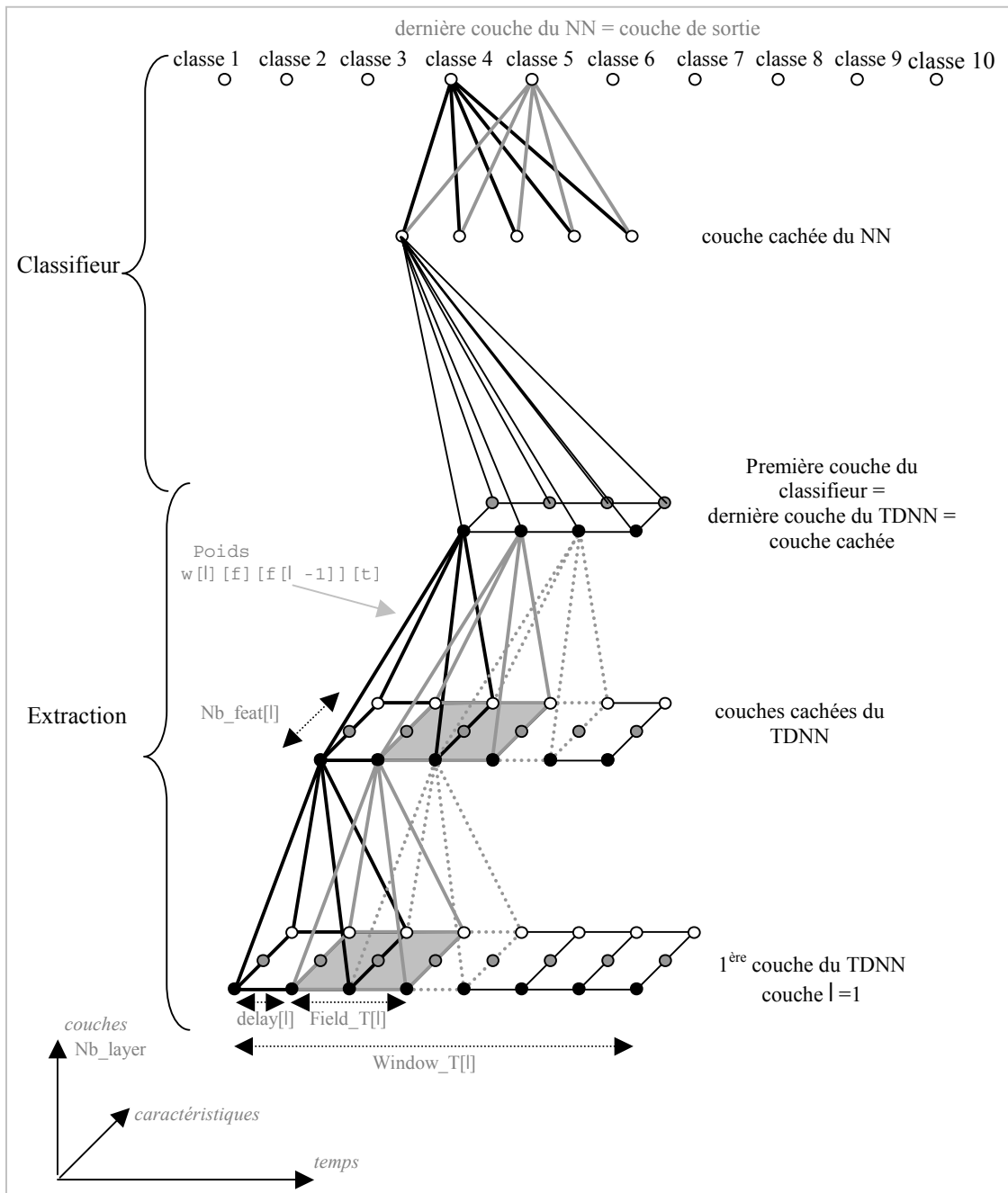


Figure 3 : Architecture du TDNN

La partie Extraction se caractérise par :

- Le nombre de couches, 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 l vu par un neurone de la couche $l+1$, $field_T$
- Le délai temporel (nombre de neurones) entre chaque fenêtre, $delay$.

Un neurone est identifié par sa couche l , sa caractéristique f , et son emplacement temporel t . Pour chaque neurone sont définis :

- une sortie, ou encore activation du neurone, $x[l][f][t]$
- une matrice des poids des entrées, $w[l][f][f[l-1]][t]$
- un vecteur de poids des biais, $w_biais[l][f]$
- 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, $delta[l][f][f[l-1]][t]$.

Et la partie Classifieur se caractérise par :

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

Un neurone du classifieur est identifié par sa couche l , et son emplacement t . Pour chaque neurone du classifieur sont définis :

- une sortie, ou encore activation du neurone, $x[l][t]$
- une matrice des poids des entrées, $w[l][t][t[l-1]]$
- la somme pondérée des entrées, $v[l][t]$
- le terme d'erreur pour la rétro propagation du gradient, $y[l][t]$
- le gradient, $delta[l][t][t[l-1]]$.

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. Un neurone donné détecte une caractéristique topologique locale de la trajectoire du stylet. 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 (nb_feat) 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.

Les opérations réalisées par une couche du TDNN sont de type convolution. Chaque neurone k de la couche $l+1$ a un noyau de taille w (nombre de neurones de la fenêtre temporelle de la couche l) * f (nombre de caractéristiques de la couche l).

La composante temporelle de la représentation du signal d'origine est éliminée au fur et à mesure en sous échantillonnant la convolution à chaque couche. Pour compenser cette perte d'informations, le nombre de caractéristiques est multiplié. Nous avons donc une architecture de type bipyramidal. Ce réseau bipyramidal convertit progressivement des informations temporelles en des informations caractéristiques.

Enfin, la première couche de la partie classifieur (MLP entièrement connecté) correspond à la dernière couche de la partie extraction.

4.2. Principe de l'apprentissage.

Un des points importants dans la mise en œuvre d'une solution à base de réseau de neurones concerne la méthode d'apprentissage [12] retenue. Idéalement, celle-ci doit permettre de converger rapidement vers le minimum global de la fonction de coût sélectionnée. Suivant la complexité du problème d'optimisation, plusieurs types de méthodes peuvent être envisagés. D'un point de vue théorique, les méthodes de second ordre [19] (Newton, Quasi-Newton, Gauss-Newton, Levenberg-Marquardt) sont considérées comme étant les plus performantes. Toutefois dans le cadre de l'application envisagée, vu le nombre de paramètres à estimer, ces méthodes conduisent à des charges de calcul et des occupations mémoires rédhibitoires. Nous avons donc dans un premier temps utilisé une méthode du premier ordre avec le classique algorithme de la rétro-propagation du gradient, toutefois la version utilisée est la version dite du « gradient stochastique » qui permet de converger souvent plus rapidement que le gradient vrai [21].

Une méthode dérivée de [16] permettant de calculer les valeurs propres du Hessien sans avoir à faire le calcul de celui-ci explicitement est à l'étude, elle permettra alors d'adapter le pas du gradient en fonction de ces valeurs, ce qui permettrait de diminuer considérablement le nombre d'itérations nécessaire tout en ne mobilisant que des ressources réduites. Rappelons que nos soucis de limitation de puissance de calcul et de capacité mémoire sont liés aux environnements mobiles (Windows CE, Palm OS) sur lesquels les algorithmes devront fonctionner, y compris par moments pour la partie apprentissage (personnalisation du système à l'utilisateur : apprentissage spécifique).

5. Les premiers résultats avec un TDNN

Le système décrit précédemment a été implémenté et mis en œuvre pour classifier des caractères isolés provenant des bases IRONOFF [23] et UNIPEN [11]. Nous avons considéré séparément les chiffres, les lettres minuscules et les lettres majuscules. Ces bases ont été acquises avec des matériels variés, 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 .

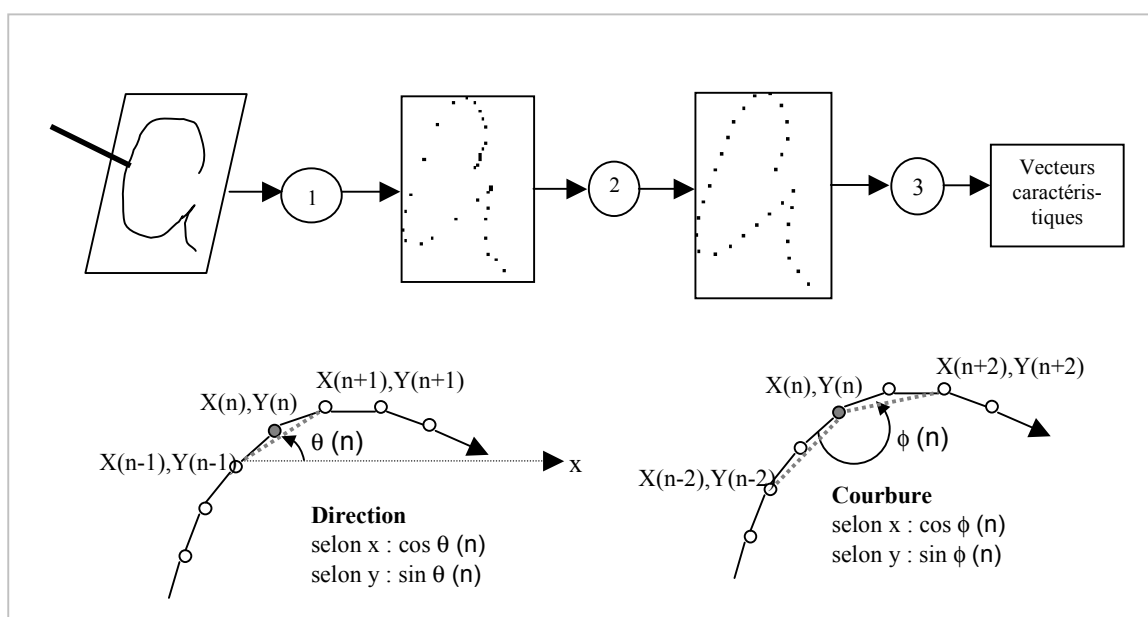


Figure 4 : Représentation du processus d'extraction en-ligne
(1 : acquisition du signal – 2 : rééchantillonnage – 3 : extraction des caractéristiques)

A partir de ces données, on ré-échantillonne spatialement ces points de façon à obtenir des points équidistants et ainsi de s'affranchir de la variabilité de la vitesse du tracé. Ainsi, nous avons fixé le

nombre de points à 50 pour chaque caractère. En chacun de ces points, un vecteur de 7 caractéristiques est extrait. Il comporte les coordonnées x et y , la direction ($\cos\theta$, $\sin\theta$) et la courbure ($\cos\Phi$, $\sin\Phi$) de la trajectoire et l'état posé ou levé du stylet en ce point, cf. figure 4.

La figure 5 illustre le caractère «a» ré-échantillonné et sa représentation intermédiaire correspondant à la séquence temporelle des vecteurs comprenant les sept caractéristiques de chaque échantillon. Le temps augmente selon l'axe horizontal de gauche à droite. Chaque vecteur est représenté par une colonne quadrillée.

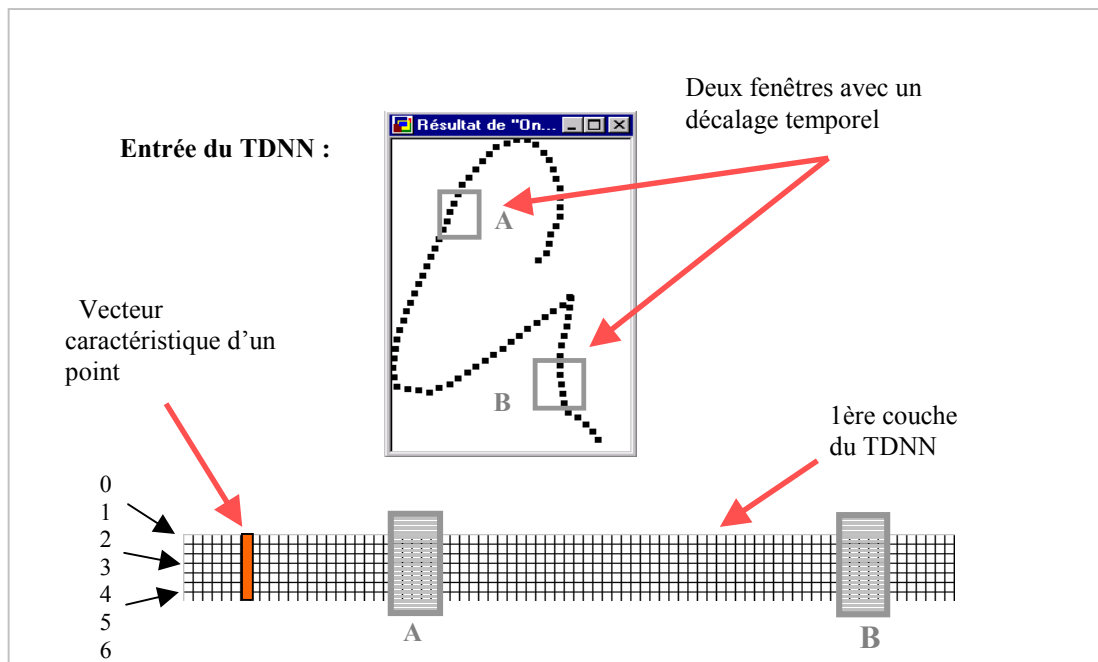


Figure 5 : Illustration du caractère 'a' à l'entrée du TDNN

Exemple d'architecture nominale testée (1 couche cachée pour le TDNN et 1 couche cachée pour le classifieur MLP) :

Couche d'entrée du TDNN à poids partagés $l = 0$:

- nombre de neurones selon la direction temporelle, $window_T[0] = 50$
- nombre de neurones selon la direction caractéristique, $nb_feat[0] = 7$
- taille du champ perceptuel (fenêtre) de la couche supérieure, $field_T[0] = 20$
- délai temporel (nombre de neurones) entre chaque fenêtre, $delay[0] = 5$

Couche cachée du TDNN, $l = 1$ (couche d'entrée du classifieur NN, $nn_l = 0$) :

- nombre de neurones selon la direction temporelle, $window_T[1] = 7$
- nombre de neurones selon la direction caractéristique, $nb_feat[1] = 20$

Couche cachée du MLP, $nn_l = 1$

- nombre de neurones $NN_nb_neuron[1] = 100$

Couche de sortie MLP, $nn_l = 2$

- nombre de neurones $NN_nb_neuron[2] =$ nombre de classes (10 pour les chiffres, 26 pour les lettres).

Les courbes présentées aux figures 6, 7 et 9 illustrent les résultats de reconnaissances obtenus sur la base de chiffres (échantillonnés sur 50 points) en s'intéressant respectivement à la sensibilité au pas du gradient, aux paramètres de convolution (taille de la fenêtre et délai temporel) à la contrainte des poids partagés en considérant l'influence du classifieur avec ou sans couche cachée par rapport à la configuration nominale donnée ci-dessus. La structure considérée du réseau est donnée pour chacun de ces exemples.

Dans le cadre de la validation du comportement du TDNN, nous avons arrêté l'algorithme de rétropropagation du gradient à un nombre fixé d'itérations de la base d'apprentissage, ici 300 itérations. Par la suite nous avons géré l'arrêt de l'algorithme selon la technique dite « early stopping », arrêt de l'algorithme lorsque la fonction de coût sur la base d'exemples de validation ayant atteint son minimum le dépasse.

Le pas du gradient est un facteur déterminant dans la vitesse de convergence du réseau de neurones (illustré figure 6). Les temps d'apprentissage croissant très vite avec la complexité du réseau, il faut trouver un pas optimal [16]. Plus le pas est petit, plus le nombre d'itérations de la base d'apprentissage sera important pour obtenir un bon taux d'apprentissage et de généralisation. Tandis que plus le pas est grand, plus ce nombre d'itérations nécessaire sera faible mais comme le montre la figure 6, le réseau risque de diverger. Dans le cadre de nos expériences, une itération de la base d'apprentissage des chiffres (13 482 exemples) dure environ 30 secondes (Pentium II-450 et utilisation des fonctions optimisées de la bibliothèque Intel).

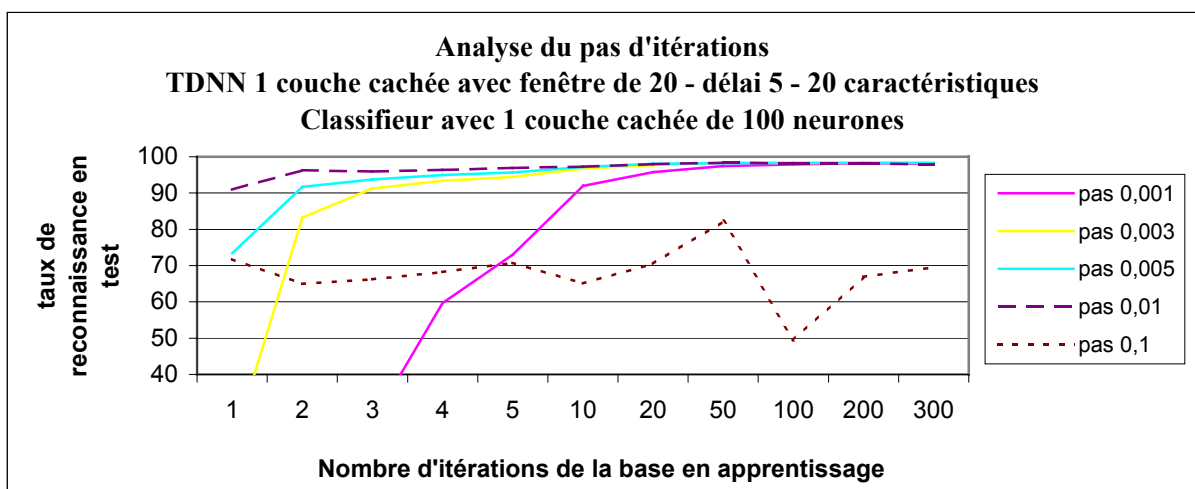


Figure 6 : taux de reconnaissance des chiffres en généralisation – Analyse du pas

Le TDNN développé est à poids partagés, ce principe de poids partagés confère au TDNN une aptitude à gérer les invariances par translations dans le temps. Le nombre de poids partagés représente le nombre de paramètres libres du réseau, il dépend de la taille de la fenêtre et du délai de chaque couche de la partie extraction du TDNN (illustré figure 8). En utilisant ainsi un délai temporel, 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.

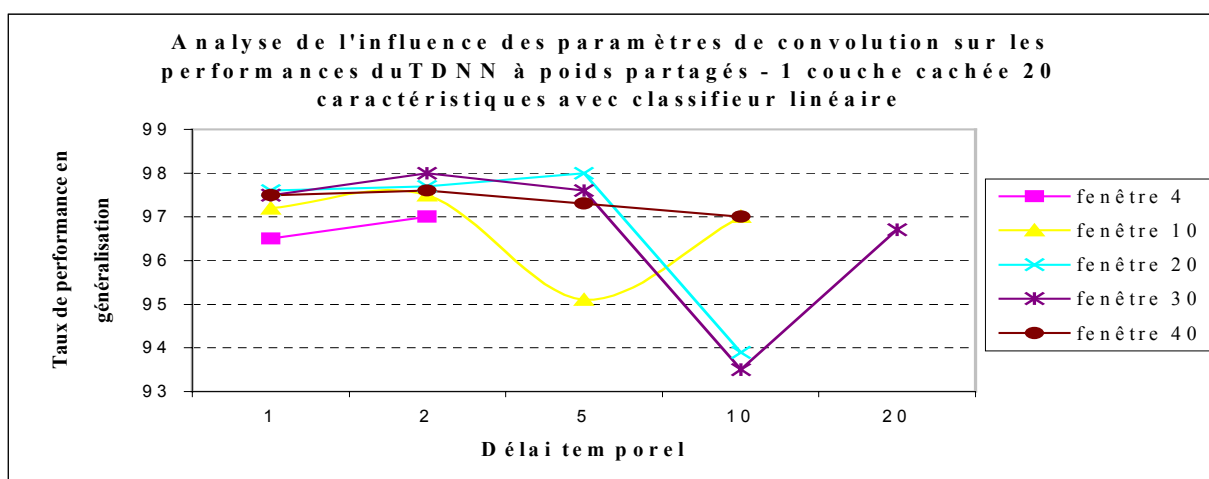


Figure 7 : taux de reconnaissance des chiffres en généralisation – Analyse de la taille de la fenêtre

Par dimensionnement de la taille de la fenêtre et du délai pour chaque couche, on influence la capacité de mémorisation du réseau. Plus on augmente la taille de la fenêtre, ou plus on diminue le délai temporel, plus le nombre de paramètres libres augmente et plus on augmente la capacité de mémorisation. Cependant une capacité de mémorisation trop importante peut nuire au pouvoir de généralisation du réseau.

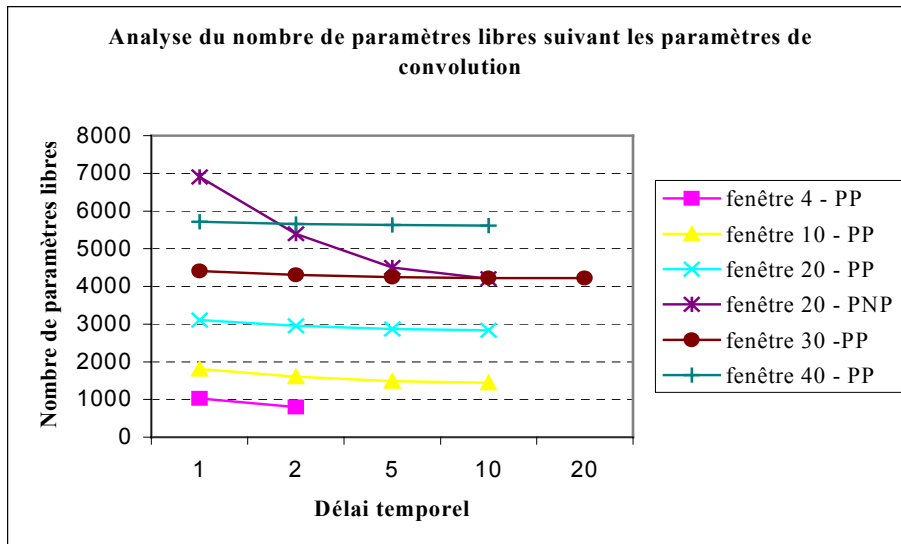


Figure 8 : Nombre de paramètres libres du TDNN suivant sa topologie pour l'apprentissage des chiffres. (PP : poids partagés, PNP : poids non partagés)

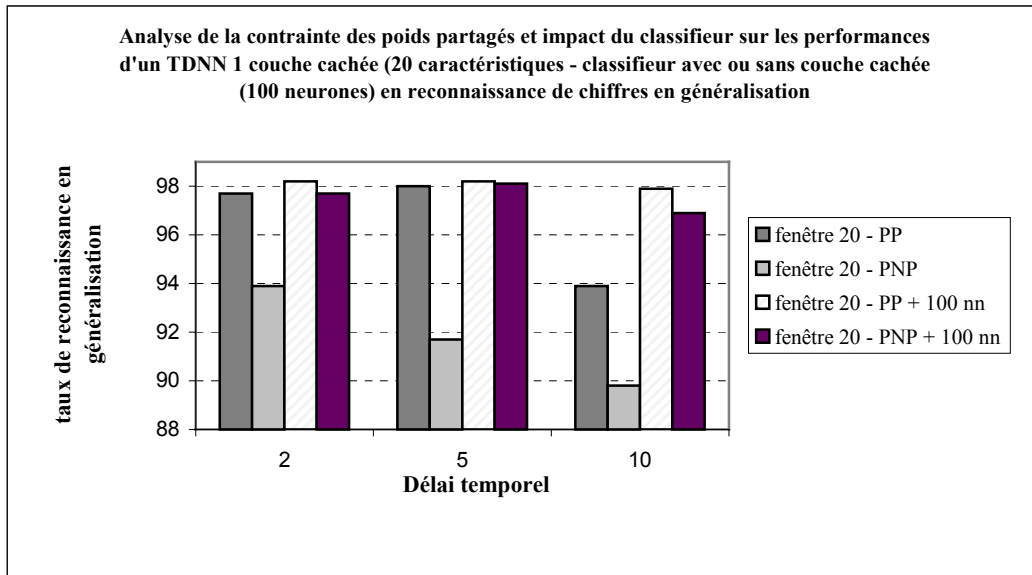


Figure 9 : Taux de reconnaissance des chiffres en généralisation – Analyse de la contrainte de poids partagés

Le tableau suivant récapitule les meilleurs résultats actuels obtenus avec le TDNN nominal à poids partagés (1 couche cachée pour la partie extraction et 1 couche cachée de 100 neurones pour le classifieur soit 17 970 paramètres libres pour les chiffres) et les résultats obtenus avec un perceptron multi-couche (1 couche cachée de 100 neurones soit 36 360 paramètres libres pour les chiffres) pour comparaison. Les caractéristiques du réseau sont celles présentées précédemment comme exemple d'architecture avec un pas gradient fixe de 0,003.

Base	Nbre de Classes	Nbre d'ex. en apprentissage	Nbre d'ex. en test	Nombre d'itérations (TDNN)	TDNN Taux de reconnaissance sur la base de test	Nombre d'itérations (MLP)	MLP Taux de reconnaissance sur la base de test
Chiffres	10	13 482	6 722	85	98.4	36	97.9
Minuscules	26	42 796	21 339	49	92.7	32	91.3
Majuscules	26	25 689	12 795	74	94.5	31	93.0

6. Conclusions et perspectives

Le travail a consisté à développer une solution à base de réseaux de neurones de type TDNN pour la reconnaissance de caractères manuscrits isolés et analyser la définition de la topologie du réseau : taille d'analyse de la fenêtre, nombre de couches, contraintes sur le partage des poids. Les premiers résultats obtenus sont prometteurs.

L'objectif de cette thèse est de contribuer à progresser dans le difficile problème de la reconnaissance de l'écriture manuscrite non contrainte. Pour cela, nous préconisons une approche s'appuyant sur des méthodes statistiques combinant des réseaux de neurones et des modèles de Markov cachés. Les premiers sont utiles pour leur fort pouvoir discriminant intrinsèque permettant de détecter des caractéristiques spécifiques (caractère par exemple, ou à un autre niveau, mots ou corps de mots, ...), les seconds ont une très bonne aptitude à modéliser les processus stochastiques tels que ceux conduisant aux séquences produites par de l'écriture manuscrite.

L'apport des travaux 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 (TDNN et SDNN). Ces réseaux sont spécifiquement destinés à traiter des formes 2D et aptes à prendre en compte leur variabilité. La mise en parallèle d'un TDNN et d'un SDNN permettrait de coupler des informations complémentaires : en-ligne et hors-ligne. La reconnaissance de caractères manuscrits issus d'une tablette digitale et d'un stylet est 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. Le scripteur ayant produit une image lisible du caractère, l'information spatiale (approche SDNN) semble importante à conserver pour préserver la nature picturale du signal. Dans le but d'une reconnaissance basée mot, l'approche de type SDNN [15], représentation spatiale, présente l'avantage de ne pas nécessiter d'heuristiques de segmentation préalables : le SDNN cherche à balayer l'entrée par une fenêtre de taille la plus proche possible d'un caractère.

Les résultats de la reconnaissance fournis par le réseau serviront à alimenter des modèles de Markov cachés. Ceux-ci sont de plus en plus utilisés en traitement d'images. Ainsi, un autre aspect de la thèse consistera à définir des modèles adaptés aux structures à reconnaître. Il faudra en effet *intégrer les caractéristiques à différents niveaux afin de modéliser correctement les effets contextuels*, et cela dans un cadre statistique. Une attention particulière sera portée sur la prise en compte d'une détection automatique du style d'écriture de telle sorte que les indices structuraux participent à la description et complètent la modélisation statistique. Une *généralisation de la programmation dynamique classique à un espace 2D adapté à l'image* doit fournir une réponse très efficace à ce problème.

On proposera également de transposer et d'évaluer systématiquement les méthodes développées à la fois pour les systèmes de reconnaissance de l'écriture statique et dynamique.

Le développement se fera par étapes. On a commencé par appliquer les méthodes proposées au problème de la reconnaissance de caractères manuscrits isolés pour lequel nous avons déjà des

solutions très performantes qui serviront de référence (MLP, TDNN). La seconde étape conduira à utiliser de tels modèles de caractères pour développer un système de reconnaissance de mots manuscrits isolés. Une autre étape, procédant par généralisation en introduisant des niveaux supplémentaires couplés de manière hiérarchique étudiera une production manuscrite globale.

Références

- [1] N. Ben Amara, A. Belaïd, N. Ellouze : « Modélisation pseudo bidimensionnelle pour la reconnaissance de chaînes de caractères arabes imprimés », in: Colloque International Francophone sur l'ECrit et le Document - CIFED'98, Québec, Canada, p. 131-141, 1998.
- [2] Y. Bengio : Cours IFT6265 « Poids Partagés, TDNNs et réseaux à convolution », Université de Montréal - Département d'informatique et de recherche opérationnelle - 1997
<http://www.iro.umontreal.ca/~bengioy/ift6265/convo/convo.html>
- [3] Y. Bengio, Y. LeCun, C. Nohl, et al. : "A NN/HMM hybrid for on-line handwriting recognition". *Neural Computation*, 7(6): pp.1289-1303, November 1995.
- [4] Y. Bengio, Y. Le Cun : "World-Level Training of a Handwritten Word Recognizer Based on Convolutional Neural Network", International Conference on Pattern Recognition, pp. 88-92, 1994.
- [5] C.M. Bishop : « Neural Networks for Pattern Recognition », *Oxford University Press*. ISBN 0-19-853849-9, pp. 116-161, 1995.
- [6] L. Bottou, C. Cortes, J.S. Denker, et al. : "Comparison of classifier methods: a case study in handwritten digit recognition." In IAPR, editor, *IEEE Proc. of the International Conference on Pattern Recognition*, volume II, pp. 77-82, Jerusalem, October 1994.
- [7] S. Garcia-Salicetti, B. Dorizzi, P. Gallinari et al. : "A Hidden Markov Model extension of a neural predictive system for on-line character recognition", ICDAR'95, *International Conference on Document Analysis and Recognition*, volume I, pp. 50-53, Montréal, 1995.
- [8] S. Garcia-Salicetti, B. Dorizzi, P. Gallinari, Z. Wimmer : "Adaptive Discrimination in an HMM-based neural predictive system for on-line word recognition", ICPR 96, *International Conference on Pattern Recognition*, 1996.
- [9] M. Gilloux : "Reconnaissance de chiffres manuscrits par modèle de Markov pseudo-2D", CNED, pp. 11-17, Rouen, 1994.
- [10] I. Guyon, P. Albrecht, I. Le Cun, et al. : "Design of a Neural Network Character Recognizer for a Touch Terminal", *Pattern Recognition*, 24 (2), 1991.
- [11] I. Guyon, L. Schomaker, S. Janet, et al. : First UNIPEN benchmark of on-line handwriting recognizers organized by NIST. Technical Report BL0113590-940630-18TM, AT&T Bell Laboratories, 1994
- [12] J. Herault, C. Jutten : Livre «Réseaux neuronaux et traitement du signal », *Edition Hermes*, p. 75-114, Juin 1994.
- [13] S. Lawrence, C. Lee Giles, Ah Chung Tsoi et al. : "Face Recognition : A Convolutional Neural Network Approach" – *IEEE Transactions on Neural Networks*, Special Issue on Neural Networks and Pattern Recognition, Volume 8, Number 1, pp. 98-113, 1997

- [14] Y. Le Cun, B. Boser, J.S. Denker, et al. : “Handwritten digit recognition with a backpropagation neural network”. In D. Touretzky editor, *Advances in Neural Information Processing Systems 2*, pp. 396-304, 1990.
- [15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998. 46 pages.
- [16] Y. LeCun, P. Simard, and B. Pearlmutter : “Automatic learning rate maximization by on-line estimation of the hessian's eigenvectors”. In S. Hanson, J. Cowan, and L. Giles, editors, *Advances in Neural Information Processing Systems*, volume 5. Morgan Kaufmann Publishers, San Mateo, CA, 1993.
- [17] S. Manke, M. Finke, A. Waibel : “Npen++ : a Writer Independant, Large Vocabulary On-Line Cursive Handwriting Recognition System”, ICDAR’95, *3th International Conference on Document Analysis and Recognition*, pp.403-408, Montréal, 1995
- [18] S. Marukatat, T. Artieres, B. Dorizzi, P. Gallinari : “Sentence Recognition through hybrid neuro-markovian modelling”, Actes de ICDAR’01, *6th International Conference on Document Analysis and Recognition*, Volume I, pp. 731-735, Seattle, 2001.
- [19] M. Minoux : Livre «Programmation Mathématique, Théorie et algorithmes » tome 1 pp 95-168 – *Collection Technique et Scientifique des Télécommunications* – ENST, édition DUNOD 1983.
- [20] L.R. Rabiner, B.H. Juang : “An Introduction to Hidden Markov Models”, *IEEE ASSP Magazine*, p4-16,1986.
- [21] W.S. Sarle, ed. (1997), *Neural Network FAQ*, part 2 : Part 2: “Learning, What are batch, incremental, on-line, off-line, deterministic, stochastic, adaptive, instantaneous, pattern, constructive, and sequential learning?”, periodic posting to the Usenet newsgroup comp.ai.neural-nets, URL : <ftp://ftp.sas.com/pub/neural/FAQ.html>
- [22] M. Schenkel, I. Guyon, D. Henderson : “On-Line Cursive Script Recognition using Time Delay Neural Networks and Hidden Markov Models”, *Machine Vision and Applications*, pp. 215-223, 1995.
- [23] C. Viard-Gaudin, P.M. Lallican, S. Knerr, P. Binter, "The IRESTE ON-OFF (IRONOFF) Handwritten Image Database", ICDAR’99, *International Conference on Document Analysis and Recognition*, pp. 455-458, Bangalore, 1999.
- [24] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K. Lang : “Phoneme recognition using time-delay neural networks”. *IEEE Transactions on Acoustics, Speech and Signal Processing*, volume 37 pp. 328-339, March 1989.