

Une représentation floue régionale pour la reconnaissance en-ligne de chiffres manuscrits

Jean-François Hébert[†], Marc Parizeau[†], Nadia Ghazzali[‡]

[†]Laboratoire de Vision et Systèmes Numériques (LVSN),
Département de génie électrique et de génie informatique

[‡]Département de mathématiques et de statistique

^{†‡}Université Laval, Ste-Foy, (Québec), G1K 7P4, Canada
e-mail: [jfhebert,parizeau]@gel.ulaval.ca, ghazzali@mat.ulaval.ca

Résumé

Nous introduisons dans cet article une nouvelle représentation floue pour décrire des caractères isolés. Cette représentation réalise la transformation de l'espace 2D des caractères vers un nouvel espace vectoriel flou. Une étape de segmentation des caractères en arcs de cercle et en segments de droite est préalable à l'application de l'algorithme que nous proposons. En dernier lieu, des expériences de reconnaissance sur des chiffres isolés extraits de la banque UNIPEN sont menées afin de comparer les performances de diverses variantes de la représentation floue. Le réseau de neurones artificiels KP est utilisé comme outil de classification.

Mots clés: Représentation floue, Réseaux de neurones, Reconnaissance de caractères.

1 Introduction

Les réseaux de neurones artificiels sont des outils reconnus comme étant de bons classifieurs [1]. Par conséquent, ils sont utilisés de plus en plus fréquemment pour résoudre des problèmes reliés à la reconnaissance d'écriture cursive [2, 3, 4]. Néanmoins, une contrainte importante qu'ils imposent est la nécessité de réaliser au préalable une transformation de l'espace des données initiales (par exemple une séquence de coordonnées dans un espace 2D) vers un espace vectoriel de dimension fixe qui contiendrait toute l'information pertinente à la reconnaissance. Diverses méthodes ont déjà été envisagées en reconnaissance d'écriture en-ligne pour la représentation des caractères [5], mais peu d'entre elles permettent d'obtenir une dimensionnalité de la représentation qui soit fixe pour tout caractère.

Dans cet article, nous proposons une nouvelle représentation vectorielle des caractères isolés. La particularité de notre approche est d'emprunter à la logique floue certains concepts nous permettant d'exprimer des régions de caractères préalablement segmentés en arcs de cercle et en segments de droite sous la forme de vecteurs flous. Comparativement à d'autres approches [2, 3], notre représentation a le mérite de s'exprimer dans un espace vectoriel de dimension relativement faible.

La suite de cet article est subdivisée comme suit. D'abord, la section 2 définit brièvement quelques notions de logique floue que nous utiliserons à la section 3 lors de la description formelle de la représentation floue proposée. Ensuite, nous présentons à la section 4 le réseau de neurones KP qui sera employé à la section 5 pour mesurer expérimentalement les performances de la représentation. Ces performances seront mesurées à l'aide de taux de reconnaissance sur des données réelles, c'est-à-dire des chiffres isolés extraits de la banque internationale UNIPEN.

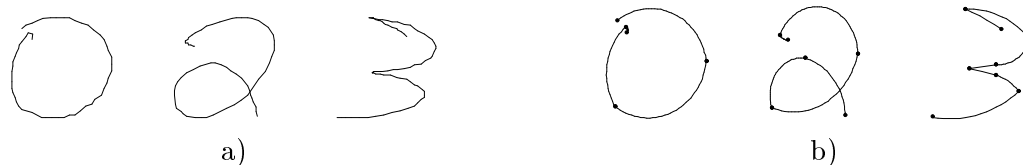


Figure 1: a) Quelques caractères d'écriture et b) leur reconstruction suite à une segmentation en arcs de cercle et en segment de droite.

2 Variables et ensembles flous

En logique floue, la “fuzzification” d’une variable X consiste à mesurer pour une observation quelconque $x \in X$ son degré d’appartenance $\mu_{\mathbf{F}}(x)$ à un ensemble flou \mathbf{F} . Ce degré d’appartenance permet de constater à quel point l’observation x appartient à l’ensemble \mathbf{F} [6]. Plus formellement, un ensemble flou \mathbf{F} est habituellement défini comme une transformation $\mu_{\mathbf{F}}(x) : X \rightarrow [0, 1]$ qui associe à chaque observation x une valeur floue (i.e. le degré d’appartenance de x à \mathbf{F}) comprise entre 0 et 1. Habituellement, plusieurs ensembles flous $\mathbf{F}_1, \mathbf{F}_2, \dots, \mathbf{F}_n$ sont définis pour une même variable X . Nous disons alors que $\langle \mu_{\mathbf{F}_1}(x), \mu_{\mathbf{F}_2}(x), \dots, \mu_{\mathbf{F}_n}(x) \rangle$ forme le vecteur flou associé à l’observation x .

3 Représentation floue

La représentation floue décrite dans cette section est réalisée en deux étapes successives. D’abord, une étape de segmentation des caractères en une séquence d’arcs de cercle et de segments de droite, puis une transformation de l’espace à dimension variable résultant de cette segmentation vers un nouvel espace vectoriel flou à dimension fixe.

3.1 Segmentation des caractères

Nous utilisons pour la segmentation des caractères une procédure récemment proposée par X. Li et al. [7] qui permet de décomposer un segment d’écriture en une séquence de traits élémentaires. Cette segmentation utilise comme hypothèse sous-jacente le modèle d’écriture delta log-normal (modèle DLN) qui permet de modéliser les mouvements rapides qui surviennent en écriture cursive [8]. La figure 1 illustre diverses séquences d’arcs de cercle résultant de l’application de cette procédure de segmentation.

Plus formellement, nous considérerons par la suite un caractère segmenté \mathcal{C} comme étant formé d’une suite de q traits élémentaires (“stroke”) $\mathcal{C} = s_1, s_2, \dots, s_q$. Un trait élémentaire représente un arc de cercle décrit par:

$$s = \langle p_0, p_1, l, c \rangle, \quad (1)$$

où $p_0 = (x_0, y_0)$ et $p_1 = (x_1, y_1)$ sont respectivement les points de départ et les points d’arrivée du trait s , l est sa longueur curviligne et c sa courbure. L’orientation θ du vecteur qui relie le point p_0 au point p_1 est un paramètre complémentaire pouvant être déduit directement des points de départ et d’arrivée. La figure 2 illustre le sens géométrique qui doit être associé à chacun de ces paramètres.

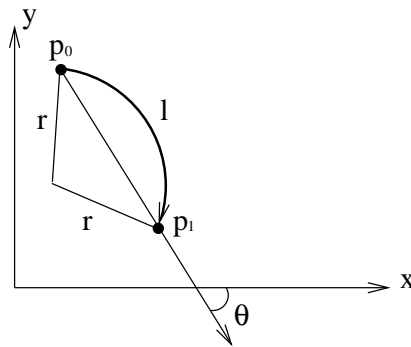


Figure 2: Modèle géométrique d'un arc de cercle. Les points p_0 et p_1 sont respectivement les points de départ et d'arrivée, l est la longueur curviligne et $r = 1/|c|$ est le rayon de courbure.

3.2 Espace vectoriel flou

Le nouvel espace vectoriel flou proposé pour la représentation des caractères est obtenu en découpant leur espace 2D en un nombre fixe de régions¹. Chacune de ces régions se verra associer un vecteur flou résultant d'un processus de fuzzification de variables sur divers ensembles flous. Le rôle de ce vecteur flou sera alors de refléter d'une manière à la fois précise et concise le contenu d'une région d'un caractère. Par conséquent un compromis devra être réalisé quant au nombre de variables qui pourront être fuzzifiées ainsi qu'au nombre d'ensembles flous qui devront être définis pour chacune de ces variables. Notons qu'une représentation trop précise peut mettre en évidence des détails inutiles pour la reconnaissance, tandis qu'à l'inverse une représentation exagérément concise peut éliminer des caractéristiques importantes qui pourraient être utiles pour discriminer certains caractères.

Expérimentalement, nous avons déterminé qu'une région particulière d'un caractère pouvait être représentée adéquatement en mesurant les degrés d'appartenance des divers traits qui la traversent à chacun des sept ensembles flous illustrés à la figure 3. Les quatre ensembles flous décrits à la figure 3a permettent de fuzzifier l'orientation $\theta \in [-180, 180]$ d'un trait, tandis que les trois ensembles de la figure 3b réalisent plutôt une fuzzification de sa courbure $c \in [-\infty, \infty]$. Ainsi, pour une région particulière, un trait pourra être considéré plus ou moins horizontal (**H**), vertical (**V**), oblique positif (**O+**) ou oblique négatif (**O-**) selon son orientation θ . Pareillement, un trait sera considéré rectiligne (**R**), courbé positivement (**C+**) ou courbé négativement (**C-**) selon sa valeur de courbure c . Notons que les recouvrements entre les divers ensembles flous font en sorte qu'un trait peut avoir des degrés d'appartenance non nuls pour deux ensembles flous distincts. Par exemple, un trait ayant une orientation telle que $15 \leq \theta \leq 30$ degrés serait considéré peu horizontal et peu oblique positif. Puisqu'un seul vecteur flou peut être associé à chaque région d'un caractère, la présence de plusieurs traits dans une région est traitée en ne considérant que les degrés d'appartenance maximaux à chacun des ensembles flous. Ceci permet ainsi de refléter la présence de plusieurs traits d'orientation ou de courbure différentes dans une même région.

La figure 4 décrit étape par étape la procédure permettant de déterminer la représentation floue d'un caractère déjà segmenté. Cet algorithme doit être considéré comme un point de départ, et divers raffinements seront proposés dans les sections subséquentes pour le perfec-

¹Notons que tous les caractères sont préalablement normalisés à une même hauteur et à une même largeur, à l'exception des caractères très étroits tel que leur ratio largeur/hauteur est inférieur à 0.25. De manière à préserver leur forme, ces derniers ne sont normalisés qu'en hauteur et sont centrés en largeur.

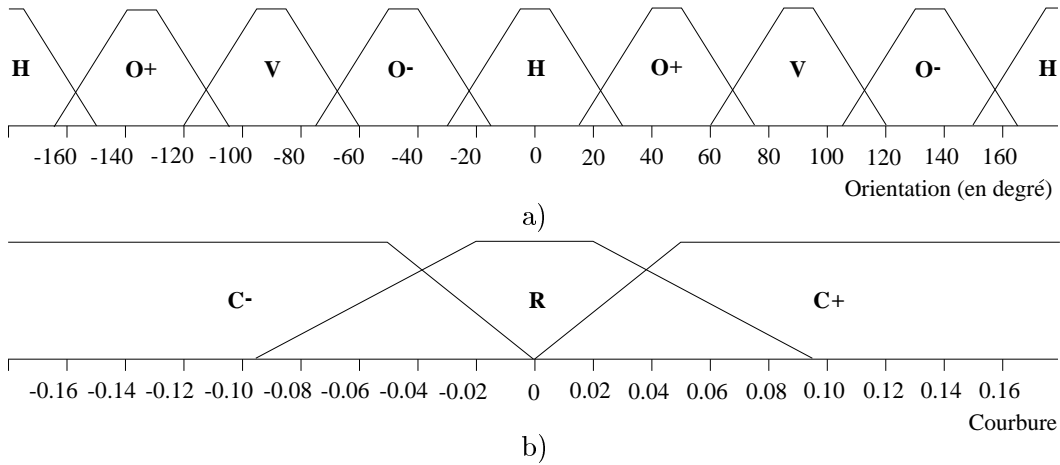


Figure 3: a) Les ensembles flous **H**, **V**, **O+** et **O-** utilisés pour la fuzzification de l'orientation $\Theta = \{\theta\}$ d'un trait et b) **R**, **C+** et **C-** pour la fuzzification de la courbure $C = \{c\}$.

Étape 1: Ajuster le caractère segmenté C à l'intérieur d'une grille rectangulaire G divisée en $n \times m$ régions (disjointes ou non).

Étape 2: Pour chaque région $g_{i,j}$ de G , $i = 1, 2, \dots, n; j = 1, 2, \dots, m$ déterminer l'ensemble $\mathbf{S}_{i,j} = \{s | s \in g_{i,j}\}$ des traits qui la croisent^a. Former pour tous les traits $s \in \mathbf{S}_{i,j}$ le vecteur flou \tilde{s} qui lui est associé:

$$\tilde{s} = \langle \mu_{\mathbf{H}}(\theta_s), \mu_{\mathbf{V}}(\theta_s), \mu_{\mathbf{O+}}(\theta_s), \mu_{\mathbf{O-}}(\theta_s), \mu_{\mathbf{R}}(c_s), \mu_{\mathbf{C+}}(c_s), \mu_{\mathbf{C-}}(c_s) \rangle$$

où θ_s et c_s sont l'orientation et la courbure de s .

Étape 3: Obtenir le vecteur flou $\tilde{g}_{i,j}$ associé à chaque région $g_{i,j}$:

$$\tilde{g}_{i,j} = \langle \nu_{\mathbf{H}}^{i,j}, \nu_{\mathbf{V}}^{i,j}, \nu_{\mathbf{O+}}^{i,j}, \nu_{\mathbf{O-}}^{i,j}, \nu_{\mathbf{R}}^{i,j}, \nu_{\mathbf{C+}}^{i,j}, \nu_{\mathbf{C-}}^{i,j} \rangle,$$

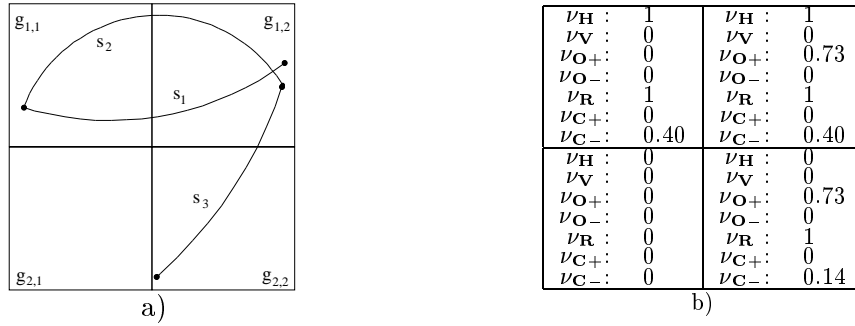
où la fonction $\nu_{\mathbf{F}}^{i,j}$ est définie pour un ensemble flou \mathbf{F} par la relation:

$$\nu_{\mathbf{F}}^{i,j} = \begin{cases} \max_{s \in \mathbf{S}_{i,j}} \mu_{\mathbf{F}}(\theta_s) & \text{si } \mathbf{F} \in \{\mathbf{H}, \mathbf{V}, \mathbf{O+}, \mathbf{O-}\} \\ \max_{s \in \mathbf{S}_{i,j}} \mu_{\mathbf{F}}(c_s) & \text{si } \mathbf{F} \in \{\mathbf{R}, \mathbf{C+}, \mathbf{C-}\} \end{cases}$$

Étape 4: Finalement, juxtaposer chacune des régions floues $\tilde{g}_{i,j}$ pour obtenir la représentation floue \tilde{C} d'un caractère segmenté C .

^aUn trait s croise une région $g_{i,j}$ (i.e. $s \in g_{i,j}$) s'il est complètement inclus dans $g_{i,j}$, ou si au moins une arête de $g_{i,j}$ est traversée par s .

Figure 4: Algorithme de représentation floue.

Figure 5: Segmentation ajustée dans une grille 2×2 et b) représentation floue de base.

tionner. Néanmoins, les modifications suggérées seront indépendantes les unes des autres et n'affecteront pas le cheminement en quatre étapes de la procédure.

Comme premier exemple, considérons le chiffre '9' segmenté en trois traits s_1, s_2 et s_3 et qui a été ajusté à l'intérieur d'une grille composée de 2×2 régions à la figure 5a. La représentation floue $\tilde{C} = \langle \tilde{g}_{1,1}, \tilde{g}_{1,2}, \tilde{g}_{2,1}, \tilde{g}_{2,2} \rangle$ obtenue à la figure 5b indique que la région $g_{2,1}$ est vide et que la région $g_{2,2}$ est traversée par un trait d'orientation oblique positif peu courbé, ce qui correspond effectivement à la partie inférieure du chiffre '9' de la figure 5a. Par contre, l'information qui est contenue dans les vecteurs flous $\tilde{g}_{1,1}$ et $\tilde{g}_{1,2}$ ne s'avère pas être idéale puisqu'elle ne met pas totalement en évidence les caractéristiques intrinsèques qui permettent de distinguer un '9' des autres chiffres. Effectivement, les vecteurs $\tilde{g}_{1,1}$ et $\tilde{g}_{1,2}$ indiquent que les régions $g_{1,1}$ et $g_{1,2}$ sont toutes deux traversées par au moins un segment horizontal ($\nu_H = 1$ pour $\tilde{g}_{1,1}$ et $\tilde{g}_{1,2}$), ce qui potentiellement correspond davantage à la description d'un '7' que d'un '9'. Cette imprécision de l'algorithme s'explique en partie par le fait que l'extraction des vecteurs flous $\tilde{g}_{i,j}$ associés à une région $g_{i,j}$ est réalisée au niveau du trait, sans jamais prendre en compte le contenu effectif de la région $g_{i,j}$.

3.3 Segmentation régionale

Suite aux résultats obtenus à la figure 5b, nous proposons dans le but d'améliorer la précision de la représentation de diviser chaque trait s_k d'un caractère segmenté $\mathcal{C} = \langle s_1, s_2, \dots, s_q \rangle$ en autant de sous-traits régionaux $s_k^{i,j}$ qu'il y a de régions croisées par s_k . Par exemple, le caractère segmenté $\mathcal{C} = \langle s_1, s_2, s_3 \rangle$ de la figure 5a s'exprimerait suite à une division en sous-traits comme $\mathcal{C}' = \langle s_1^{1,2}, s_1^{1,1}, s_2^{1,1}, s_2^{1,2}, s_3^{1,2}, s_3^{2,2} \rangle$. Formellement, ce sont les points d'intersection entre un trait s et les arêtes d'une région $g_{i,j}$, lorsqu'ils existent, qui délimitent pour le sous-trait $s^{i,j}$ soit un point de départ $p_0^{i,j}$ si s pénètre dans la région $g_{i,j}$ ou soit un point d'arrivée $p_1^{i,j}$ s'il en sort. Notons que $p_0^{i,j} = p_0$ lorsque le trait s débute son tracé dans la région $g_{i,j}$ et que $p_1^{i,j} = p_1$ lorsque son point d'arrivée se situe dans la région $g_{i,j}$. Nous représentons un sous-trait de la même façon que nous l'avons fait à l'équation (1) pour un trait, c'est-à-dire sous la forme d'un quadruplet $s^{i,j} = \langle p_0^{i,j}, p_1^{i,j}, l^{i,j}, c \rangle$. Notez bien que cette nouvelle segmentation n'influence aucunement la courbure des traits. De plus, elle n'affecte que peu la procédure décrite à la figure 4 pour l'obtention de la représentation floue. La seule adaptation nécessaire concerne l'étape 3 où la fonction $\nu_{\mathbf{F}}^{i,j}$ ne doit plus être appliquée sur tous les traits $s \in \mathbf{S}_{i,j}$, mais plutôt sur tous les sous-traits $s^{i,j} \in \mathbf{S}_{i,j}$.

En ce qui concerne le chiffre '9' illustré à la figure 6a, sa division en sous-traits régionaux fournit la représentation floue décrite à la figure 6b. Les valeurs d'orientations floues ν_H, ν_V, ν_{O+} et ν_{O-} associées aux deux régions $g_{1,1}$ et $g_{1,2}$ indiquent plus explicitement la présence

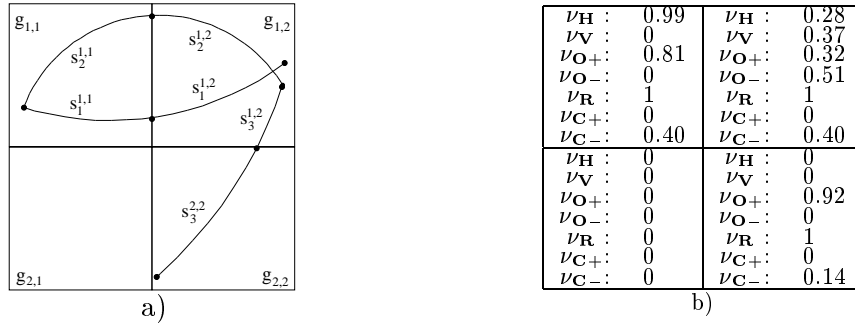


Figure 6: Segmentation en sous-traites régionaux et b) représentation floue résultante.

d'une boucle qui caractérise la partie supérieure d'un chiffre '9' que les mêmes valeurs obtenues précédemment à la figure 5b avec l'algorithme de base.

3.4 Normalisation des longueurs curvilignes

Dans les sections précédentes, la même importance a été accordé à chaque trait² pour le calcul des différents vecteurs flous d'une représentation. Or, ceci implique d'accorder autant d'importance à des traits courts qu'à des traits longs, même si ces derniers fournissent habituellement de l'information plus pertinente quant à la description d'un caractère. Ceci découle directement du fait que les variables d'orientation et de courbure qui sont employées pour la fuzzification ne tiennent compte en aucun cas d'une notion de longueur ou de norme d'un trait.

Une solution envisageable serait de définir de nouveaux ensembles flous pour prendre en compte la longueur curviligne d'un trait. Toutefois, cette solution implique une augmentation considérable de la dimensionnalité de la représentation floue³, ce qui va à l'encontre de notre volonté de concevoir une représentation concise. Pour contourner ce problème, nous suggérons plutôt d'appliquer à chaque vecteur flou \tilde{s} associé à un trait s un facteur de poids tel que:

$$\tilde{s}' = \left(\frac{l_s}{l_{\max}} \right) \tilde{s}, \quad (2)$$

où l_s est la longueur curviligne du trait s et l_{\max} est un facteur de normalisation adéquat. Dans la pratique, nous fixons l_{\max} comme étant égal à la longueur de la diagonale des régions $g_{i,j}$ de la grille G .

3.5 Densités relatives

Les densités relatives horizontales d_i^H et verticales d_j^V d'un caractère segmenté sont définies pour la $i^{\text{ème}}$ ligne et la $j^{\text{ème}}$ colonne d'une grille G comme suit:

$$d_i^H = \frac{1}{L} \sum_{j=1}^m \sum_{s_k^{i,j} \in \mathbf{S}_{i,j}} l_k^{i,j}, \quad i = 1, 2, \dots, n; \quad d_j^V = \frac{1}{L} \sum_{i=1}^n \sum_{s_k^{i,j} \in \mathbf{S}_{i,j}} l_k^{i,j}, \quad j = 1, 2, \dots, m, \quad (3)$$

où $l_k^{i,j}$ est la longueur curviligne du trait $s_k^{i,j}$ et où $L = \sum_{i=1}^q l_i$ est la longueur curviligne totale du caractère segmenté $\mathcal{C} = \langle s_1, s_2, \dots, s_q \rangle$. Ces densités relatives peuvent être intégrées au

²Rappelons qu'un sous-trait est également un trait.

³Pour le cas d'une grille G de dimension $n \times m$ (voir la figure 4), chaque addition d'un ensemble flou augmente de $n \times m$ la dimensionnalité de la représentation.

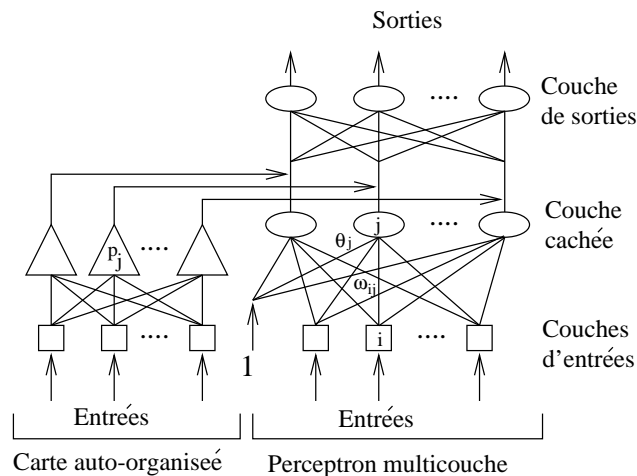


Figure 7: Architecture du réseau KP.

sein de la représentation floue afin de permettre une analyse de plus haut niveau qui tendrait à préciser la description d'un caractère. Selon les besoins, différentes autres caractéristiques de plus haut niveau, tel que par exemple le ratio hauteur sur largeur d'un caractère avant normalisation, peuvent être également être considérées.

4 Réseau de neurones KP

Pour tester les performances de notre représentation floue, nous réaliserons diverses expériences de reconnaissance sur des chiffres isolés. Le réseau de neurones qui sera utilisé pour la classification de ces chiffres est le réseau KP conçu par Guillot et Azouzi [9]. Nous avons préféré utiliser ce nouveau réseau artificiel plutôt qu'un réseau perceptron multicouche (PMC) [11] principalement car le réseau KP possède une architecture hybride qui lui confère des possibilités intéressantes d'apprentissage incrémentales [10].

Comme son nom l'indique, un réseau KP consiste en la juxtaposition d'une carte d'auto-organisation de Kohonen [12] (réseau K) et d'un réseau PMC (réseau P). Le lien entre ces deux réseaux distincts survient au niveau de la dernière couche cachée du réseau P, chaque neurone de cette couche étant connecté à un neurone du réseau K. Par conséquent, le nombre de neurones sur la dernière couche cachée du réseau P doit correspondre exactement au nombre total de neurones du réseau K, tel qu'illustré à la figure 7.

L'objectif visé par cette combinaison de réseaux est de pouvoir profiter des capacités bien connues de modélisation des cartes auto-organisées pour déterminer quelle région (i.e. quel groupe de neurones) du réseau P doit contribuer davantage à l'émission du vecteur des sorties. En mode de reconnaissance, une donnée est présentée simultanément aux deux réseaux préalablement entraînés: le réseau K détermine pour chacun de ses neurones une valeur de contribution, puis transmet cette information au réseau P qui l'utilise comme un facteur de pondération supplémentaire (en plus du poids de ses neurones) pour déterminer l'activation de chaque neurone à sa sortie.

En mode d'apprentissage, les réseaux P et K sont entraînés séparément. Les algorithmes d'apprentissage utilisés sont respectivement l'algorithme non-supervisé de Kohonen [12] pour le réseau K et l'algorithme de rétropropagation des poids pour le réseau P [11]. Il est nécessaire de débiter par l'apprentissage du réseau K puisque celui-ci doit calculer les contributions neu-

ronales de chaque entrée qui doivent être utilisées par le réseau P à la fois pour propager une entrée à travers le réseau et pour rétropropager les erreurs obtenues à la sortie. L'algorithme de rétropropagation des erreurs doit être adapté pour tenir compte des contributions neuronales lors de l'adaptation des poids du réseau P [9].

5 Reconnaissance de chiffres isolés

Diverses expériences de reconnaissance ont été menées sur des chiffres isolés (0-9) extraits de la banque internationale UNIPEN [13]. Les chiffres contenus dans la version Train-R01/V07 (15953 chiffres) ont constitué la banque d'apprentissage, alors que les chiffres de la version DevTest-R02/V02 (8555 chiffres) ont formé la banque de test. Afin de fournir des résultats qui soient facilement comparables d'une étude à l'autre, tous les chiffres extraits de la banque ont été conservés, y compris les chiffres de très mauvaise qualité⁴.

Le tableau 1 énonce des taux de reconnaissance sur la banque de test pour diverses variantes des représentations floues, en plus d'une représentation selon une grille binaire qui consiste à diviser chaque chiffre en une grille de $n \times m$ pixels et d'associer à chaque pixel la valeur 1 si le tracé croise le pixel ou la valeur 0 dans le cas contraire. La représentation "Base" est obtenue à l'aide de la procédure de base décrite à la figure 4, tandis que les représentations "Base+Reg." et "Base+Reg.+Long." tiennent comptes respectivement d'une segmentation régionale et d'une normalisation des longueurs curvilignes. La représentation "Finale" est la plus complète de toutes puisqu'elle contient à la fois les informations de densités relatives, les ratios hauteur/largeur et largeur/hauteur du caractère, en plus d'une normalisation des longueurs curvilignes et d'une segmentation régionale. La structure des réseaux KP utilisés pour toutes les expériences de reconnaissance est la même, soit une carte auto-organisée de Kohonen de 15×15 neurones imbriquée à l'intérieur d'un réseau PMC de structure $k \times (15 \times 15) \times 10$, c'est-à-dire un réseau de k entrées (k étant la dimensionnalité de la représentation floue), (15×15) neurones sur une couche cachée et 10 neurones en sorties. Cette structure a démontré expérimentalement les meilleurs résultats. Les vecteurs de sorties désirées pour chaque chiffre de la banque d'apprentissage ont été déterminés en assignant à chacune des dix classes de chiffre un neurone de sortie particulier. En mode d'apprentissage, chaque exemplaire présenté doit chercher à activer fortement (0.99) le neurone de sortie associé à sa classe d'appartenance et chercher à inhiber autant que possible (0.01) les autres neurones de sortie. Soulignons que dans tous les cas, les réseaux KP ont utilisé 1000000 itérations⁵ pour entraîner la carte auto-organisée. Le nombre de passes⁶ d'entraînement du réseau PMC varie quant à lui d'une expérience à l'autre dans le but d'éviter le surapprentissage du réseau. À cet effet, l'entraînement du PMC se terminait lorsque la racine de l'erreur quadratique moyenne commençait à stagner.

Suite à l'entraînement du réseau KP sur les chiffres de la banque d'apprentissage, chaque exemplaire de la banque de test est reconnu selon la meilleure hypothèse (1-hyp) si l'argument maximal de son vecteur de sorties observées correspond à l'argument maximal de son vecteur des sorties désirées. Lorsque les arguments maximaux des vecteurs de sorties observées et désirées coïncident plutôt au niveau de la deuxième ou de la troisième valeur maximale, nous dirons que le chiffre est reconnu en deuxième ou troisième hypothèse (2-hyp ou 3-hyp).

Les taux de reconnaissance donnés au tableau 1 mettent en évidence l'intérêt que nous

⁴Seuls 43 chiffres de la banque de test ont été éliminés, leurs dimensions en X et en Y étant de 0×0 !

⁵Une itération correspond à la présentation d'une donnée choisie aléatoirement.

⁶Une passe d'entraînement correspond pour le PMC à la présentation une fois de toutes les données d'entraînement.

Tableau 1: Taux de reconnaissance des chiffres de la banque de test pour différentes représentations obtenues selon différentes dimensions de grille. Les taux sont énumérés pour la meilleure hypothèse (1-hyp) ainsi que pour les deux et trois meilleures hypothèses (2-hyp et 3-hyp). La dimensionnalité de la représentation (Dim.), le nombre de passes d'entraînement du réseau PMC (Pa.) et la racine de l'erreur quadratique moyenne obtenue (REQM) sont également spécifiés dans le tableau.

Représentation	Grille	Dim.	Pa.	REQM	1-hyp.	2-hyp.	3-hyp.
Binaire	8×6	48	400	0.073	89.46%	94.67%	96.27%
	10×8	80	400	0.067	89.14%	94.15%	95.81%
	14×10	140	300	0.061	88.76%	93.52%	95.46%
Base	3×2	42	300	0.070	90.60%	95.69%	97.16%
Base+Reg.	3×2	42	400	0.059	91.61%	95.98%	97.27%
Base+Reg.+Long.	3×2	42	300	0.051	94.58%	97.31%	98.08%
Finale	3×2	49	600	0.044	95.25%	97.87%	98.59%
	4×3	93	300	0.037	96.10%	98.16%	99.10%
	5×4	151	300	0.038	96.29%	98.39%	99.02%

avons à combiner différents niveaux d'analyse pour la représentation floue d'un chiffre. En effet, concernant la représentation "Finale" et une grille de dimension 3×2, le taux de reconnaissance obtenu selon la meilleure hypothèse est supérieur de plus de 4.5% au taux obtenu avec la représentation "Base", et de pratiquement 6% au taux obtenu avec la représentation "Binaire" pour une grille de dimension 8×6. De plus, nous considérons d'une manière absolue que l'obtention d'un taux de reconnaissance de 95.25% selon la meilleure hypothèse est un excellent résultat car nous estimons⁷ qu'au moins 25% des chiffres non reconnus par le réseau KP seraient difficilement reconnaissables par un humain, bon nombre d'entre eux étant carrément méconnaissables soit parce que mal étiqueté, soit à cause d'une incompatibilité dans les échelles (i.e. axe X en pouce et axe Y en centimètre), ou soit parce que segmenté en plusieurs parties (e.g. les deux composantes d'un '4' séparées en 2 caractères distincts, tous les deux étant étiquetés comme des '4'). Si l'on fait l'hypothèse conservatrice que tous les chiffres reconnus par le réseau le seraient aussi par un humain, nous pouvons supposer que le taux de reconnaissance d'un humain sur la banque de test serait approximativement de 98.81% (i.e. $95.25\% + 0.75 \times (100\% - 95.25\%)$). Notons que ce plafond n'est supérieur que de 0.22% au taux obtenu par le réseau KP selon les trois meilleures hypothèses.

Le tableau 1 énonce également des taux de reconnaissance pour des représentations de dimensionnalité variable. Les résultats démontrent que le fait d'augmenter la résolution de la représentation "Finale" permet d'accroître les taux de reconnaissance, alors que c'est l'effet inverse qui se produit pour la représentation "Binaire".

6 Conclusion

Nous avons introduit dans la première partie de cet article un algorithme général pour l'obtention d'une nouvelle représentation floue de caractères isolés. Nous avons par la suite défini divers raffinements pouvant être apportés à l'algorithme dans le but de préciser la représentation floue. Dans la dernière partie, nous avons démontré à l'aide d'expériences de reconnaissance

⁷Cette estimation résulte d'une expérience auprès de 10 personnes à qui l'on a demandé de classer les chiffres non reconnus par notre réseau. Le taux d'erreur moyen a été de 27.3%.

sur des chiffres isolés que la représentation proposée combinée à un classifieur de type réseau KP produisait des taux de reconnaissance compétitifs.

7 Remerciements

Cette recherche a été rendue possible en partie grâce à des subventions CRSNG accordées à M. Parizeau et N. Ghazzali et grâce à une subvention FCAR accordée à M. Parizeau.

Références

- [1] L. Holmström, P. Koistinen, J. Laaksonen, E. Oja, "Neural and Statistical Classifiers - Taxonomy and Two Case Studies", *IEEE Trans. on Neural Networks*, vol. 8, no. 1, pp. 5-17, 1997
- [2] I. Guyon, P. Albrecht, Y. LeCun, J. Denker, W. Hubbard, "Design of a Neural Network Character Recognizer for a Touch Terminal", *Pattern Recognition*, vol. 24, no. 2, pp. 105-119, 1991
- [3] Y. Bengio, Y. LeCun, C. Nohl, C. Burges, "LeRec: A NN/HMM Hybrid for On-Line Handwriting Recognition", *Neural Computation*, vol. 7, no. 5, 1995
- [4] L.R.B. Schomaker, "Using Stroke- or Characted-based Self-Organizing Maps in the Recognition of On-Line, Connected Cursive Script", *Pattern Recognition*, vol. 26, no. 3, pp. 443-450, 1993
- [5] C.C. Tappert, C.Y. Suen, T. Wakahara, "The State of the Art in On-Line Handwriting Recognition", *IEEE Trans. on Pattern and Machine Intelligence*, vol. 12, no. 8, pp. 787-808, 1990
- [6] S.V. Kartalopoulos, *Understanding Neural Networks and Fuzzy Logic*, IEEE Press, 1996
- [7] X. Li, M. Parizeau, R. Plamondon, "Segmentation and Reconstruction of On-Line Handwritten Scripts", *Pattern Recognition*, vol. 31, no. 6, 1998
- [8] R. Plamondon, "Handwriting Generation: the Delta Log-Normal Theory", *Proc. of the Fourth International Workshop on Frontiers in Handwriting Recognition*, pp. 1-10, 1994
- [9] M. Guillot, R. Azouzi, "Improving On-Line Adaptation in Neurocontrol using a Combination of Self-organizing Map and Multilayer Feedforward Network", apparait dans *Intelligent Engineering Systems Through Artificial Neural Networks*, édité par C.H. Dagli, B.R. Fernandez, J. Ghosh, S. Kumara, vol. 4, ASME Press, pp. 915-922, 1995
- [10] J.-F. Hébert, M. Parizeau, N. Ghazzali, "Incremental supervised learning using a NGCS-P neural network", soumis à *International Conference on Pattern Recognition*, Brisbane, Australie, 1998
- [11] S. Haykin, *Neural Networks: A Comprehensive Foundation*, IEEE Press, 1994
- [12] T. Kohonen, "The Self-Organizing Map", *Proc. of the IEEE*, vol. 78, no. 9, pp. 1464-1480, 1990
- [13] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, S. Janet, "UNIPEN Project of On-Line Data Exchange and Recognizer Benchmarks", *Proceedings of the 12th International Conference on Pattern Recognition*, pp. 29-33, Jerusalem, Israel, 1994