

Reconnaissance en-ligne de caractères arabes manuscrits par un réseau de Kohonen

Neila Mezghani¹, Amar Mitiche¹, Mohamed Cheriet²,

1: INRS-Télécommunications, Université du Québec

2: Ecole de Technologie Supérieure, Université du Québec

Neila@inrs-telecom.quebec.ca

Abstracts

Neural networks have been applied to various pattern classification and recognition problems for their learning ability, discrimination power, and generalization ability. The neural networks most referenced in the pattern recognition literature are the multi-layer perceptron, the Kohonen associative memory and the Carpenter-Grossberg ART network. The Kohonen memory runs an unsupervised clustering algorithm. It is easily trained and has attractive properties such as topological ordering and good generalization. In this study an on-line system for the recognition of handwritten Arabic characters using a Kohonen network is investigated. The input of the neural network is a feature vector of elliptic Fourier coefficients extracted from the handwritten dynamic representation. Experimental results show that the network successfully recognizes both clearly and roughly written characters with superior performance.

1 Introduction

Bien que les premières recherches sur la reconnaissance en-ligne de caractères arabes soient réalisées au début des années 80, peu de travaux ont été menés par la suite [1, 2, 3]. Dans [1], l'auteur divise les caractères arabes en deux catégories: les caractères stables et les caractères instables. Il adopte une représentation syntaxique particulière pour chaque catégorie. La classification est basée sur l'appariement des chaînes. Dans [2], les caractères sont divisés en un groupe de caractères indépendant du scripteur et un autre dépendant de celui-ci. Les vecteurs de caractéristiques sont formés par les codes de Freeman. Une structure d'arbre est utilisée pour la classification. Dans [3], les caractères sont représentées par une suite de nombre de directions et de positions et la reconnaissance s'appuie sur la programmation dynamique.

Toutes ces méthodes permettent d'obtenir de bons résultats sur le compte d'une complexité calculatoire (e.g, pro-

grammation dynamique). De même, l'utilisation de la représentation syntaxique présente les inconvénients d'être sensibles aux distorsions (problème couramment rencontré pour l'écriture manuscrite) et de ne pas avoir un bon pouvoir de généralisation.

Le but de ce travail est de développer une mémoire de Kohonen pour la reconnaissance en-ligne de caractères arabes manuscrits. La mémoire de Kohonen s'obtient par un apprentissage non supervisé, elle présente des propriétés attractives telles que l'ordonnement et le pouvoir de généralisation [4, 5, 6]. Les caractéristiques sont extraites à partir de la représentation dynamique des caractères à travers les signaux $x(t)$ et $y(t)$, qui correspondent à la position du stylo électronique sur la tablette au temps t .

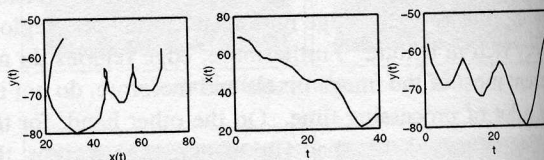


FIG. 1 – Représentation dynamique du caractère "Seen"

Cet article est divisé en cinq sections. La deuxième section présente le réseau de Kohonen et son algorithme d'évolution. La troisième section contient la représentation adoptée des caractères. Dans la quatrième section, nous donnons et interprèterons les résultats obtenus dans les différentes expériences réalisées. Une conclusion et quelques perspectives sont données dans la dernière section.

2 Le réseau de Kohonen

Le réseau de Kohonen est composé de deux couches; la première constitue l'entrée du réseau et la seconde est la sortie. Le réseau réalise une liaison entre une entrée à I composantes et un ensemble de J sorties par l'intermédiaire de

$I \times J$ coefficients aussi appelés poids (Fig. 2).

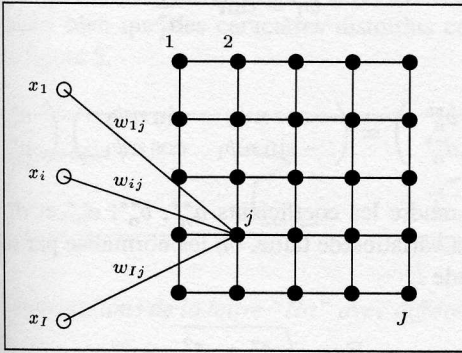


FIG. 2 – Réseau de Kohonen

L'espace d'entrée E est de dimension I , une entrée étant un vecteur $X = (x_1, x_2, \dots, x_I)^T$. Chaque composante x_i est reliée aux J noeuds de sortie, ou neurones, par J coefficients W_{ij} . Chaque sortie j peut être donc considérée comme porteuse d'un vecteur image $W_j = (W_{1j}, W_{2j}, \dots, W_{Ij})^t$.

La carte topologique de Kohonen (*Self-Organizing Map*) impose une structuration spéciale à ses neurones. Cette structuration lie les neurones et les contraint à respecter une certaine topologie lors de l'apprentissage. Ainsi, des données proches dans l'espace d'entrée ont des représentations proches dans la carte topologique de Kohonen.

Algorithme d'évolution de la mémoire de Kohonen

L'algorithme d'évolution d'auto-organisation de la mémoire de Kohonen peut se présenter ainsi:

- *Initialisation des poids*: initialiser (itération 0) les poids W_j^0 à de petites valeurs aléatoires, $j \in [1, J]$.
- *Présentation des données*: présenter une donnée en entrée à l'itération n courante, $X^n = (x_1^n, \dots, x_I^n)^T$, et calculer sa distance à chacun des vecteurs $W_j^n, j \in [1, J]$.
- *Selection du noeud gagnant*: sélectionner le noeud j^* le plus proche de l'entrée en calculant les distances

$$d_j^2 = \sum_{i=1}^I (x_i(t) - W_{ij}(t))^2 \quad (1)$$

- *Mise à jour*: mettre à jour les poids, selon la formule 2

$$W_{ij}^{n+1} = W_{ij}^n + \epsilon_n h_n^{j,j^*} (x^n - W_{ij}^n) \quad (2)$$

$$i \in [1, I], j \in [1, J]$$

$$\epsilon_n = \epsilon_i \left(\frac{\epsilon_f}{\epsilon_i} \right)^{\frac{n}{n_{max}}} \quad (3)$$

$$h_n^{j,j^*} = \exp - \frac{\|j - j^*\|^2}{2\sigma_n^2} \quad (4)$$

$$\sigma_n = \sigma_i \left(\frac{\sigma_f}{\sigma_i} \right)^{\frac{n}{n_{max}}} \quad (5)$$

Dans ces expressions:

- W_{ij}^n représente le poids entre l'entrée i et la sortie j , à l'itération n .
- Le noeud j^* est celui qui, à l'itération n , est le plus proche de l'entrée $X^n = (x_1^n, \dots, x_I^n)^T$.
- La fonction h_n^{j,j^*} définit l'influence du point j^* sur le point j ; elle décroît avec la distance entre les positions j et j^* de ces noeuds dans la grille de sortie. La grille que nous avons utilisée est de forme carrée.
- ϵ_n est un facteur affectant l'amplitude des changements apportés aux poids à l'itération n .
- Les paramètres initiaux ϵ_i et σ_i et finaux ϵ_f et σ_f doivent être fixés de manière à ce que le système converge, ce qui comprend en fait deux phases:
 1. Ordonnancement des poids W_{ij} .
 2. Convergence des vecteurs poids vers leur valeur finale.

Pendant la phase d'entraînement, un vecteur $X = (x_1, \dots, x_I)^T$ est présenté au J noeuds de la mémoire contenant les vecteurs $W_j = (w_{1j}, \dots, w_{Ij})^T$ initialisés à des valeurs aléatoires. Les vecteurs des poids de la mémoire sont tous mis à jours selon la relation (2). La mise à jour des voisins dépend de la fonction d'étendue h^{j,j^*} . Le résultat obtenu après la phase d'entraînement est une mémoire contenant un ensemble de vecteurs de poids $W_j = (w_{1j}, \dots, w_{Ij})^T$ affectés aux noeuds $j, j \in [1, J]$.

Pendant la phase de classification, on présente un vecteur inconnu $X = (x_1, \dots, x_I)^T$ et on détermine le noeud le plus proche en terme de distance. Le vecteur X portera l'étiquette de la classe du noeud gagnant.

3 Représentation

La représentation des caractères sous forme de caractéristiques joue un rôle important dans le processus de reconnaissance. Nous avons adopté les descripteurs elliptiques de Fourier [7] sur des contours fermés et nous les avons utilisées sur les signaux $x(t)$ et $y(t)$ de l'écriture en-ligne.

L'approximation des signaux $(x(t), y(t), t = 1 \dots m)$ est donnée par:

$$\hat{x}(t) = A_0 + \sum_{n=1}^N \left[a_n \cos \frac{2n\pi t}{T} + b_n \sin \frac{2n\pi t}{T} \right] \quad (6)$$

$$\hat{y}(t) = C_0 + \sum_{n=1}^N \left[c_n \cos \frac{2n\pi t}{T} + d_n \sin \frac{2n\pi t}{T} \right] \quad (7)$$

où T est la longueur totale du signal, et $\hat{x}(t) \equiv x(t)$ et $\hat{y}(t) \equiv y(t)$ quand $N \rightarrow \infty$.

Les coefficients a_n , b_n , c_n et d_n sont calculés selon:

$$a_n = \frac{T}{2n^2\pi^2} \sum_{i=1}^m \frac{\Delta x_i}{\Delta t_i} [\cos \phi_i - \cos \phi_{i-1}] \quad (8)$$

$$b_n = \frac{T}{2n^2\pi^2} \sum_{i=1}^m \frac{\Delta x_i}{\Delta t_i} [\sin \phi_i - \sin \phi_{i-1}] \quad (9)$$

$$c_n = \frac{T}{2n^2\pi^2} \sum_{i=1}^m \frac{\Delta y_i}{\Delta t_i} [\cos \phi_i - \cos \phi_{i-1}] \quad (10)$$

$$d_n = \frac{T}{2n^2\pi^2} \sum_{i=1}^m \frac{\Delta y_i}{\Delta t_i} [\sin \phi_i - \sin \phi_{i-1}] \quad (11)$$

avec:

$$\phi_i = \frac{2n\pi t_i}{T} \quad (12)$$

$$\Delta x_i = x_i - x_{i-1}, \quad \Delta y_i = y_i - y_{i-1} \quad (13)$$

$$\Delta t_i = \sqrt{\Delta x^2 + \Delta y^2} \quad (14)$$

$$t_i = \sum_{j=1}^i \Delta t_j \quad (15)$$

$$T = t_m = \sum_{j=1}^m \Delta t_j \quad (16)$$

x_i et y_i représentent la position du stylo (abscisse et ordonnée) sur la tablette à l'instant t_i , m est le nombre des points du caractère, T est la durée totale d'écriture et n est l'ordre des descripteurs.

Pour rendre ces coefficients indépendants du choix du point de départ, on calcule le décalage par rapport au premier axe majeur

$$\theta_1 = \frac{1}{2} \tan^{-1} \frac{2(a_1 b_1 + c_1 d_1)}{a_1^2 - b_1^2 + c_1^2 - d_1^2} \quad (17)$$

et on fait une rotation selon (eq. 18) pour rendre le décalage nul,

$$\begin{pmatrix} a_n^* & b_n^* \\ c_n^* & d_n^* \end{pmatrix} = \begin{pmatrix} a_n & b_n \\ c_n & d_n \end{pmatrix} \begin{pmatrix} \cos n\theta_1 & -\sin n\theta_1 \\ \sin n\theta_1 & \cos n\theta_1 \end{pmatrix} \quad (18)$$

De même, les coefficients sont rendus invariants à la rotation en calculant l'angle par rapport au premier axe majeur.

$$\psi_1 = \tan^{-1} \frac{c_1^*}{a_1^*} \quad (19)$$

$$\begin{pmatrix} a_n^{**} & b_n^{**} \\ c_n^{**} & d_n^{**} \end{pmatrix} = \begin{pmatrix} \cos n\psi_1 & \sin n\psi_1 \\ -\sin n\psi_1 & \cos n\psi_1 \end{pmatrix} \begin{pmatrix} a_n^* & b_n^* \\ c_n^* & d_n^* \end{pmatrix} \quad (20)$$

Pour rendre les coefficients a_n^{**} , b_n^{**} , c_n^{**} et d_n^{**} , invariants à la variation de taille, on les normalise par rapport à l'amplitude E

$$E = \sqrt{a_1^{*2} + c_1^{*2}} = a_1^{**} \quad (21)$$

La rotation utilisée dans l'équation (20) donne $b_1^{**} = c_1^{**} = 0.0$, ces coefficients ne sont donc pas utilisés comme attributs. De même, a_1^{**} étant une combinaison linéaire de a_1^{*2} et c_1^{*2} , il n'est pas aussi utilisé.

4 Résultats expérimentaux

4.1 Description de la base de donnée

Les caractères arabes isolés sont au nombre de 29. Si on ignore les diacritiques qui n'apportent aucune information pour la reconnaissance, on obtient 18 caractères.

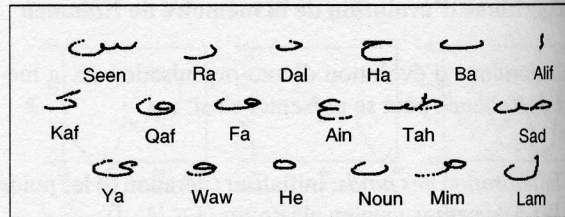


FIG. 3 – Formes des lettres choisies

Ces caractères ont été écrits 24 fois par 17 scripteurs ce qui fait un total d'environ 7400 échantillons. La figure 3 montre un exemple des caractères considérés. La base de données représente une diversité de forme et de taille. La figure 4 représente un ensemble d'échantillons du caractère "Ha".

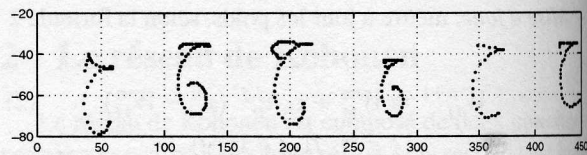


FIG. 4 – La lettre "Ha" écrites par différents scripteurs

Aucune contrainte n'a été imposée aux scripteurs; de ce fait, la base de données comprend des caractères écrits clairement aussi bien que des caractères distordus comme le montre la figure 5.

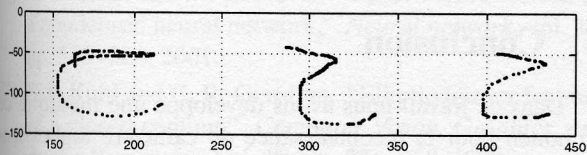


FIG. 5 – Echantillons de la lettre “Ha” avec différentes distorsions

Il est à remarquer que la donnée initiale est un vecteur contenant une succession de coordonnées $(x(t), y(t))$ et que par conséquent les vecteurs n'ont pas la même dimension, laquelle dépend du scripteur et de sa vitesse d'écriture (Fig.6)

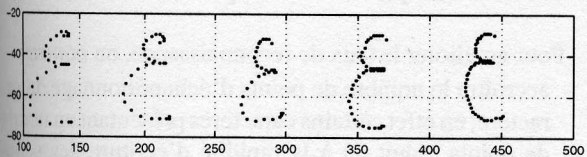


FIG. 6 – La lettre “Ain” écrite à différentes vitesses

4.2 Tests de reconnaissance

Pour étudier l'effet de la variation de certains paramètres sur la mémoire de Kohonen, nous avons exécuté trois expériences en faisant varier un paramètre à la fois.

1. la dimension du vecteur de caractéristiques;
2. le nombre d'itérations dans l'entraînement de la mémoire;
3. le nombre de noeud de la mémoire.

La base de caractères est divisée en deux ensembles, un ensemble d'entraînement contenant 5000 échantillons et une base de tests de reconnaissance comprenant 2400 échantillons.

4.2.1 Etude de la variation de la dimension du vecteur des caractéristiques

Le but de cette expérience est d'étudier la variation du taux de reconnaissance en fonction de la dimension du vecteur de caractéristiques; nous avons fixé la taille de la mémoire à 1600 et le nombre d'itérations 100.

La figure (7) montre que le meilleur taux de reconnaissance est obtenu avec 17 caractéristiques ce qui correspond

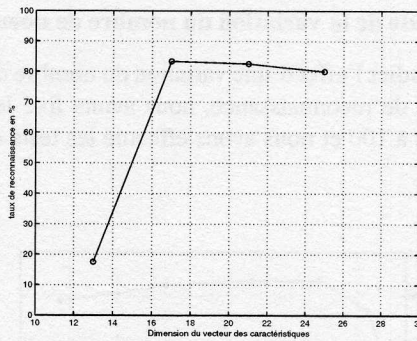


FIG. 7 – Taux de reconnaissance en fonction de la dimension du vecteur des caractéristiques

à un ordre $n = 5$ (en ignorant les coefficients a_1, b_1 et c_1). En plus, on peut remarquer qu'à partir de cette dimension le taux de reconnaissance décroît. La dimension 17 a été donc retenue.

4.2.2 Etude de la variation du nombre d'itérations

Pour l'étude de la variation du taux de reconnaissance en fonction du nombre d'itérations, nous avons fixé la taille de la mémoire à 1600 noeuds et le nombre des caractéristiques à 17.

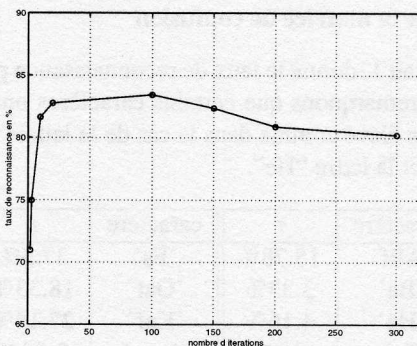


FIG. 8 – Taux de reconnaissance en fonction du nombre d'itérations

La première remarque à faire en se référant à la figure (8) est que la variation du taux de reconnaissance est rapide au début, puis après un certain nombre d'itérations, environ 80, cette variation devient lente et le taux de reconnaissance tend vers une valeur seuil de l'ordre de 83.43%. Nous constatons d'autre part qu'à partir d'environ 100 itérations, le taux de reconnaissance diminue, ce qui peut indiquer un sur-apprentissage de la mémoire.

4.2.3 Etude de la variation du nombre de noeuds

Pour étudier l'effet d'une variation du nombre de noeuds sur le taux de reconnaissance, nous avons fixé le nombre d'itérations à 100 et nous avons effectué les tests de reconnaissance.

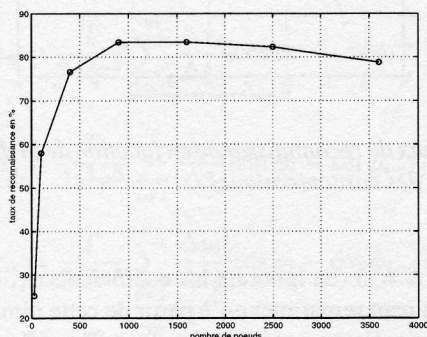


FIG. 9 – Taux de reconnaissance en fonction du nombre des noeuds

Nous constatons (Fig.9) que le taux de reconnaissance augmente en fonction du nombre de noeuds, puis à partir d'une certaine taille de la mémoire, ce taux diminue. Le taux de reconnaissance maximum est atteint pour une mémoire de taille 1600 noeuds.

4.2.4 Analyse des taux de reconnaissance par caractère et de la matrice de confusion

Le tableau 1, donne le taux de reconnaissance par caractère. Nous remarquons que certains caractères ont un taux d'erreur très élevé comme dans le cas de la lettres "Del", la lettre "Fa" et la lettre "He".

caractère	τ	caractère	τ
'Alif'	15.78%	'Fa'	37.5%
'Ba'	3.33%	'Qaf'	18.33%
'Ha'	4.16%	'Kaf'	27.14%
'Del'	40%	'Lam'	2.08%
'Ra'	33.33%	'Mim'	22.91%
'Seen'	4.16%	'Noun'	26.38%
'Sad'	5%	'He'	34.02%
'Tah'	9.16%	'Waw'	15.97%
'Ain'	3.33%	'Ya'	13.88%

TAB. 1 – Taux de reconnaissance par caractère

Pour ces caractères nous avons calculé la matrice de confusion. La majorité des erreurs provient de la grande similarité morphologique des caractères, comme par exemple la lettres

"Del" a un taux d'erreur de 40%. Dans 18.33% des cas, il est confondu avec la lettre "Noun". Pour ces caractères nous devons prévoir un traitement spécifique tel que l'usage d'un réseau de Kohonen ciblé sur ces caractères.

5 Conclusion

Dans ce travail nous avons développé une mémoire de Kohonen pour la reconnaissance de caractère arabes manuscrits en-ligne. Une mémoire de Kohonen de 1600 permet d'obtenir un taux de reconnaissance en temps réel de 84.43%.

La mémoire de Kohonen présente deux avantages. D'une part elle permet de reconnaître des données incertaines ou bruitées. D'autre part, bien que le temps d'entraînement soit long, le temps de classification est faible. L'opération d'entraînement est simple et ne se fait qu'une seule fois.

Pour améliorer le taux de reconnaissance, on pourrait:

- accroître le nombre de points d'échantillonnage du caractère; en effet certains caractères présentant un nombre de points réduit dû à la rapidité d'écriture des scribes,
- introduire des pré-traitement pour améliorer la forme du caractère et réduire les erreurs du aux tremblement de la main et à la pause du stylo,
- augmenter le nombre des données dans la base.
- pour certains caractères dont le taux de reconnaissance par caractère est faible, on pourrait déterminer un seuil minimal au dessous duquel nous faisons un traitement spécifique au caractère. Ce traitement pourrait être un entraînement en cascade de la mémoire de Kohonen pour les caractères en question.

Références

- [1] A. Amin, A. Kaced, J.-P. Haton, and R. Mohr, "Handwritten arabic character recognition by the irac system," in *Proc. Int. Conf. on Pattern Recognition*, Miami, Florida, USA, 1980, pp. 729–731.
- [2] M. S. El-wakil and A. A. Shoukry, "On-line recognition of handwritten arabic character recognition," *Pattern Recognition*, vol. 22, no. 2, pp. 97–105, 1989.
- [3] A. M. Alimi, "A neuro-fuzzy approach to recognize on-line arabic handwriting," in *Proc. of Int. Conf. on Neural Network*, Aug 1997, vol. 3, pp. 1397–1400.
- [4] A. Mitiche and J.K. Aggarwal, "Pattern category assignment by neural networks and the nearest neighbors rule," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 10, pp. 393–408, 1996.

- [5] M. Sabourin and A. Mitiche, "Modeling and classification of shape using a khonen memory with selective resolution," *Neural Networks*, vol. 6, no. 2, pp. 275–283, 1993.
- [6] A. Mitiche and M. Lebidoff, "Pattern classification by a condensed neural network," *Neural network*, vol. 14, pp. 575–580, 2001.
- [7] F. P. Kuhl and C. R. Giardina, "Elliptic fourier features of a closed contour," *Computer Vision, Graphics, and Image Processing*, vol. 18, pp. 236–258, 1982.