



## LE PAVAGE RECOUVRANT POUR LA CONSTRUCTION DU PERCEPTRON MULTI-COUCHE

C. MOLINA, P. BAYLOU, M. NAJIM

Equipe Signal et Image et GDR 134 - CNRS

ENSERB. Université de Bordeaux I

351, Cours de la libération. 33405 Talence cedex. FRANCE

Tel. 56 84 61 40      Fax. 56 84 84 06

### RÉSUMÉ

Nous présentons une nouvelle méthode neuronale, appelée *Pavage Recouvrant*, destinée à la construction du perceptron multi-couche en reconnaissance de formes. L'apport principal de ce travail est l'apprentissage et la construction du réseau neuromimétique en temps linéaire prédictible. Le *Pavage Recouvrant* dérive du *Pavage de Voronoï*, utilisé en géométrie algorithmique. Nous évaluons les performances du classifieur issu du *Pavage Recouvrant* sur la base de données "Iris" et le problème des "2-spirales".

### ABSTRACT

This paper deals with a new neural learning rule, called Overlaped Tessellation, to determine the architecture for a three-layer neural Perceptron in pattern classification. The main contribution of this work is to train and design the neural network in a predictable linear time as opposed to the unpredictability NP-time of classical methods. This method is derived from the Voronoï Tessellation algorithm which is mainly used in computational geometry. To demonstrate the Overlaped Tessellation method's efficiency we have tested it using the "Iris" and the "2-spirals" data.

## 1. INTRODUCTION

Le *Pavage de Voronoï (PV)* [1] a été récemment utilisé pour la construction des réseaux perceptron multi-couche [2]. Cette méthode permet en un temps polynomial d'en établir automatiquement l'architecture et de déterminer les poids synaptiques correspondants. Cependant, ce temps de calcul  $O(n[(d+1)/2])$ , dépendant du nombre  $n$  d'échantillons pour l'apprentissage et de la dimension  $d$  des vecteurs d'entrée, s'avère excessif en pratique [3]. La plus grande partie du temps est employée pour le calcul des facettes du **PV**. Certaines facettes discriminent des objets appartenant à des classes différentes ce qui, en termes de classification, est essentiel ; les autres discriminent des objets appartenant à la même classe et sont donc inutiles.

Dans le but de réduire la complexité calculatoire, nous proposons une règle neuronale alternative, appelée Pavage Recouvrant (**PR**). Celui-ci calcule uniquement les facettes discriminant les objets de

classes différentes et fournit à l'issue de l'apprentissage un perceptron à trois couches. Nous appliquons ensuite la méthode **PR** à la classification de la base de données "Iris" [2] et au problème des "2-spirales".

## 2. LE PAVAGE RECOUVRANT

### 2.1. Algorithme

Soit  $C = \{C_1, C_2, \dots, C_l, \dots, C_m\}$  un ensemble de classes d'objets considérés comme des points dans l'espace euclidien  $E^d$ . Pour une classe donnée  $C_l$ , le **PR** divise l'espace euclidien  $E^d$  en régions convexes de décision, appelées polytopes, contenant uniquement des objets appartenant à la classe  $C_l$ .

Soit  $p$  et  $q$  deux objets quelconques dans  $E^d$ , la facette  $H_{pq}$  est l'hyperplan perpendiculaire médian au segment  $pq$ .



*Remarque-* L'unité de temps choisie, afin d'établir l'ordre de complexité calculatoire du **PR**, est égale au temps nécessaire à la construction d'une facette.

### Algorithme Pavage Recouvrant

Soit  $np_l$  le nombre d'objets appartenant à la classe  $C_l$  et  $np_l^*$  le reste des objets, tel que  $np = np_l + np_l^*$ .

**1ère étape. Calcul du polytope  $O(np_l^*)$  :** Soit  $p$  un objet de la classe  $C_l$ , considéré comme un point dans l'espace  $d$ -dimensionnel ; construire les segments de droites partant du point  $p$  vers tous les points appartenant aux classes opposées, et calculer toutes les facettes correspondantes. Le domaine d'intersection des demi-espaces définis par les facettes délimite exactement le polytope de Voronoï issu de  $p$ .

**2ème étape. Optimisation du nombre de facettes  $O(np_l \cdot np_l^* + np_l^* \cdot \lg np_l^*)$  :** La validité des facettes est testée dans l'ordre croissant de leur distance au point  $p$ . Une facette discriminant  $p$  est inutile si, une fois éliminée, les objets appartenant aux autres classes restent exclus du nouveau domaine [4].

**3ème étape. Première optimisation des polytopes  $O(np_l)$  :** Les objets contenus dans le polytope calculé à partir de  $p$  et appartenant à la classe  $C_l$ , ne peuvent pas générer de nouveaux polytopes et doivent donc être exclus.

**4ème étape. Deuxième optimisation des polytopes  $O(np_l)$  :** Nous vérifions l'inclusion des polytopes déjà calculés  $P = \{P_1, P_2, \dots, P_k, \dots, P_n\}$  dans le nouveau polytope optimisé  $P_{n+1}$ .

Si  $\exists P_k, k=1, \dots, n$  tel que  $\forall p^i \in P_k, i=1, \dots, np_l, p^i \in P_{n+1}$ , alors le polytope  $P_k \subset P_{n+1}$  et doit être éliminé.

**5ème étape. Test d'arrêt  $O(np_l)$  :** La construction du PR se termine lorsque tous les objets de la classe  $C_l$  sont inclus dans les polytopes générés.

Soit  $n_{pol}$  le nombre de polytopes résultant de l'algorithme après  $n_{pol}$  itérations, tel que  $1 \leq n_{pol} \leq np_l$ . Le **PR** fournit un perceptron multi-couche en temps linéaire  $O(n_{pol} \cdot (np_l^* \cdot (1 + np_l + \lg np_l^*) + 3 \cdot np_l))$ .

## 2.2. Construction du réseau neuromimétique

**1ère Couche cachée.** La première couche cachée reçoit comme entrée les  $d$  paramètres réels de l'objet  $p$  à classifier. Chaque facette du polytope  $k$  correspond à un neurone de la première couche cachée et ses coefficients  $\alpha_i, i=1, \dots, d$  et  $\beta$  correspondent aux poids synaptiques des neurones. L'entrée du neurone  $j$ , appartenant au polytope  $k$ , dans la première couche est:

$$H_j^k = \sum_{i=1}^d \alpha_i \cdot p_i - \beta \quad (1)$$

où  $p_i$  est le  $i$ -ème paramètre de l'objet d'entrée  $p$  et  $\beta$  le seuil de la fonction  $H_j^k$ .

La sortie du neurone  $j$  est  $O_j^k = 1$  si  $H_j^k \geq 0$  et sinon  $O_j^k = 0$ .

**2ème Couche cachée.** Chaque polytope  $k$  de la classe  $l$  correspond à un neurone de la deuxième couche cachée et calcule le **Et** logique des sorties binaires provenant de la première couche cachée :

$$O_k^l = Et_{j=1}^{h_k} \{O_j^k\} \quad (2)$$

où  $h_k$  est le nombre de facettes du polytope  $k$ .

**Couche de sortie.** Chaque neurone de la couche de sortie correspond à une des  $m$  classes possibles et calcule le **Ou** logique des sorties binaires  $O_k^l$  provenant de la deuxième couche cachée :

$$O_l = Ou_{k=1}^{h_l} \{O_k^l\} \quad (3)$$



où  $h_l$  est le nombre de polytopes discriminant la classe  $l$ .

Nous obtenons ainsi un réseau neuromimétique pour chaque classe, dont l'architecture et les poids synaptiques sont établis automatiquement (figure 1).

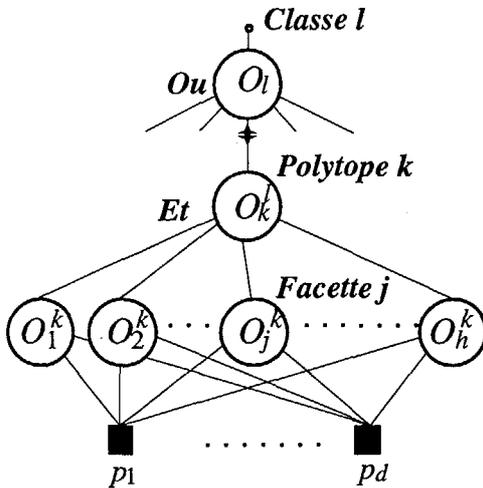


Figure 1. Réseau neuromimétique discriminant la classe  $l$ .

Les performances du réseau s'améliorent en remplaçant les fonctions binaires logiques *Et* et *Ou*, par les fonctions de la logique floue *min* et *max* [6]. Ainsi, nous obtenons une valeur continue qui représente le réel degré d'appartenance de l'objet  $p$  à sa classe. Etant donné que l'équation  $O_j$  est normalisée, la valeur issue du neurone  $O_j$  est égale à la distance euclidienne de l'objet  $p_i$  à la facette  $H_j$ ; en conséquence, la valeur  $O_l$  est la distance minimale de  $p_i$  à la frontière délimitant sa classe.

### 3. RESULTATS EXPERIMENTAUX

#### 3.1. Base de données "Iris"

La base de données "Iris" [7] contient trois classes de 50 objets, où chaque classe correspond à un type de fleur Iris: *Setosa*, *Versicolor* et *Virginica*. Chaque objet est caractérisé par quatre paramètres analogiques. Le tableau 1 présente les performances en généralisation obtenues par le **PR** comparées à celles

obtenues par huit autres méthodes [7]. La première colonne indique l'erreur apparente commise (*Err app*) par un classifieur issu des 150 objets ; la seconde colonne montre l'erreur moyenne obtenue par la technique "*laisse un de côté*" (*Err luc*) : apprentissage sur 149 objets, test d'erreur sur l'exemple restant.

METHODES	Err app	Err <sub>luc</sub>
Linéaire	0.020	0.020
Quadratique	0.020	0.027
Bayes	0.047	0.067
Bayes 2 <sup>ème</sup> ordre	0.040	0.160
CART	0.040	0.047
Rétropropagation	0.017	0.033
PVM	0.027	0.040
Pavage de Voronoï	0.000	0.040
Pavage Recouvrant	0.000	0.040

Tableau 1. Etude comparative des performances obtenues pour la base de données "Iris".

#### 3.2. Le problème des "2-spirales"

Dans le problème des "2-spirales", les méthodes de classification doivent distinguer deux spirales imbriquées, chacune formée par 96 échantillons. La figure 2 montre les régions de décision générées par le **PV** et le **PR**, ainsi que le nombre d'arêtes et de polygones (neurones de la première et de la deuxième couche cachée respectivement).

### 4. DISCUSSION

Les résultats obtenus par le **PR** révèlent des performances en généralisation comparables à celles des autres méthodes. De plus, il apporte une amélioration par rapport au **PV** en diminuant le nombre de neurones nécessaires à la résolution du problème des "2-spirales". Les caractéristiques principales du **PR** sont :



- 1) Calcul en temps linéaire d'une architecture de type perceptron à trois couches.
- 2) Construction du perceptron avec zéro erreur de classification pour l'ensemble d'apprentissage, quelle que soit la distribution des objets dans l'espace  $d$ -dimensionnel.
- 3) Optimisation du nombre de neurones des deux couches cachées durant la construction du perceptron.
- 4) Valeur continue, fournie à la sortie du perceptron, égale au degré d'appartenance de l'objet à sa propre classe.

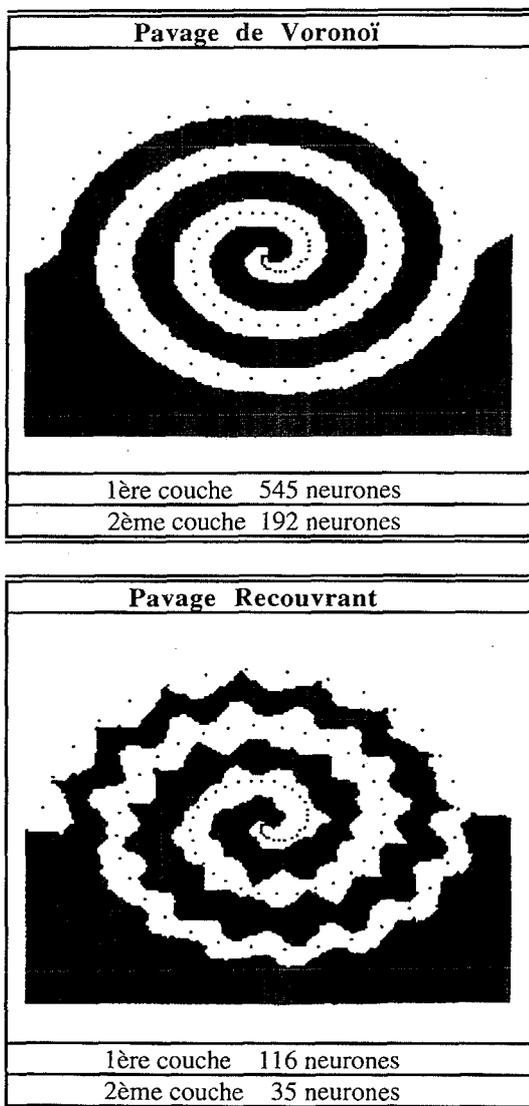


Figure 2. Résultat obtenu pour le problème des "2-spirales".

## 4. CONCLUSION

Nous avons présenté une méthode neuronale alternative pour la classification. Cette méthode est basée sur la géométrie algorithmique et montre que la construction d'un réseau multicouche est possible en temps linéaire. De plus, le réseau fourni à l'issue de l'apprentissage se prête facilement à l'interprétation grâce aux régions définies par les polytopes convexes et aux règles de la logique floue de la deuxième et troisième couche. Ces résultats encourageants nous amènent à étudier le comportement du *Pavage Recouvrant* sur des bases d'apprentissage bruitées à l'aide de techniques propres à la géométrie algorithmique.

## REFERENCES

- [1] F. P. Preparata et Shamos, "Computational Geometry," Springer-Verlag, Berlin, 1985.
- [2] Owen J. Murphy, Octobre 1990, "Nearest Neighbor Pattern Classification Perceptrons," IEEE proceedings, p. 1595-1598.
- [3] Philippe Martin et C. Bellissant, 1991, "Geometrical learning in a network of automata," Artificial Neural Networks, eds T. Kohonen & al, North-Holland, p. 1793-1796.
- [4] Philippe Martin, Avril 1992, "Réseaux de Neurones artificiels : Application à la Reconnaissance Optique de Partitions Musicales," Thèse de Doctorat de l'Université JOSEPH FOURIER. Grenoble I.
- [5] C. Molina, P. Baylou et al, "New approach for the multi-perceptron architecture construction applied to the edge detection problem," p. 1219-1222, ÉUSIPCO 92, 24-27 Août 1992, Brussels-Belgium.
- [6] C. Molina, P. Baylou et al, 1993, "Evaluation of the 'Gift-Wrapping' Neural Network in pattern classification," (à paraître) APII-Automatique, Productique, Informatique Industrielle, Edition Hermès, n. 2, 1993.
- [7] Sh. M. Weiss et I. Kapouleas, 1990, "An Empirical Comparison of Pattern Recognition, Neural Nets and Machine Learning Classification Methods," Machine Learning, p. 781-787.