

Mise en oeuvre d'un réseau de neurones simple : le Perceptron

Mathieu Petit, Thomas Le Bras

26 mars 2008

Résumé

L'un des problèmes majeurs en intelligence artificielle est la distinction entre deux classes. Il se pose comme la base de systèmes d'IA beaucoup plus complexes. Par exemple, en traitement automatique du langage, on cherche à distinguer les voyelles des consonnes; en reconnaissance des formes, on distingue un carré d'un cercle; pour les systèmes experts, il s'agit de reconnaître un cas particulier d'un autre. L'utilisation d'un réseau de neurones permet de résoudre ces problèmes de classification. Un neurone sait faire la différence entre deux classes dès lors qu'il a appris par des exemples successifs à correctement classer (apprentissage supervisé). Ce TD vous amène à découvrir les réseaux de neurones et à construire un perceptron pour le cas concret de la distinction entre deux groupes de bateaux.

Pour ce TD, nous vous demandons de nous rendre trois choses :

- le fichier source perceptron.cpp **commenté** ;
- vos notes de compte-rendu, dans lesquelles vous formulez les réponses aux questions, indiquez les résultats de vos expérimentations et les difficultés rencontrées ;
- la feuille de résultats sur laquelle vous tracerez les droites séparatrices (n'oubliez pas d'indiquer votre nom!).

1 Neurone artificiel

D'inspiration biologique, les neurones artificiels sont des petites unités de traitement composés d'une ou plusieurs entrées, d'une sortie et d'un corps de cellule qui effectue des calculs à partir des données de l'entrée pour produire la sortie (Fig 1). En pratique, un neurone est capable de discriminer deux classes, c'est à dire de produire 1 ou -1 à sa sortie.

1.1 Entrées

x_1 à x_n : données en entrée du neurone. Celles-ci peuvent être des données d'entrée du réseau ou des valeurs intermédiaires provenant de sorties d'autres neurones. Un vecteur $E_{entree}^T = (x_1, \dots, x_i, \dots, x_n)$ caractérise un élément d'entrée du neurone. Il peut s'agir par exemple de coordonnées cartésiennes $(x, y, z)^T$

Chacune de ces entrées est pondérée par un poids ω . Ces poids constituent un vecteur de pondération des entrées $P_{entree}^T = (\omega_1, \dots, \omega_i, \dots, \omega_n)$

La dernière entrée est appelée entrée de biais. Sa valeur est toujours égale à 1, son poids associé b est appelé le biais du neurone.

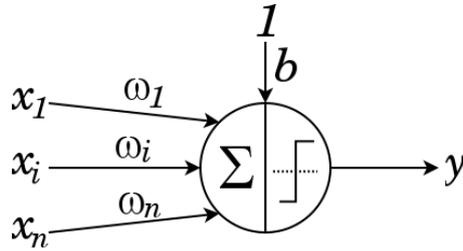


FIG. 1 – Structure d'un neurone

1.2 Cellule

L'unique fonction de la cellule est de produire une sortie égale à 1 ou -1 . Pour cela, elle effectue la somme pondérée des éléments entrés :

$$f(s) = b + \sum_{i=1}^n x_i \omega_i$$

Connaissant cette somme, on utilise une fonction de seuil qui renvoie -1 si la somme est négative ou 1 si la somme est positive :

$$y = \begin{cases} 1 & \text{si } f(s) \geq 0 \\ -1 & \text{sinon} \end{cases}$$

1.3 Sortie

La sortie d'une cellule est une valeur binaire 1 ou -1 dans le cas d'un neurone à seuil. C'est la réponse du neurone à un problème de tri entre deux ensembles d'éléments de dimension n . Dans un réseau de neurones, cette sortie (pondérée ou non) peut constituer l'entrée d'un nouveau neurone (Fig 2).

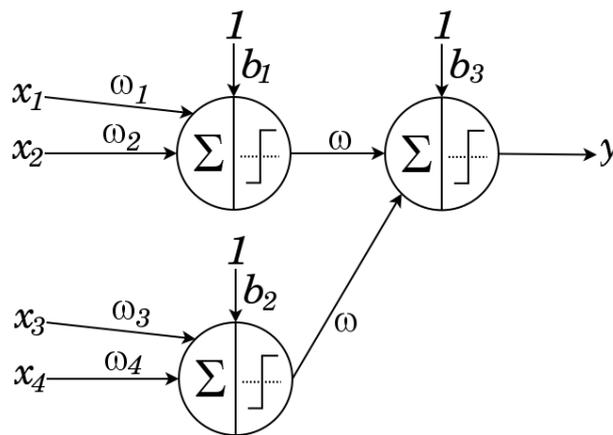


FIG. 2 – Réseau de deux neurones d'entrée et un neurone de sortie

2 Apprentissage et Classement

Nous nous intéresserons pour la suite au plus simple des réseaux de neurones : le perceptron. Inventé en 1957 par Franck Rosenblatt du laboratoire d'aéronautique de Cornell, le perceptron peut être vu comme un unique neurone capable de séparer linéairement un ensemble de vecteurs E_{entree} en deux groupes distincts étiquetés A et B

2.1 Apprentissage

Dans son état initial, un perceptron ne sais pas comment séparer les deux groupes, il faut qu'il "apprenne". Nous disposons pour cela d'un échantillon d'apprentissage constitué de vecteurs d'entrée particuliers, dont on connaît le groupe g d'appartenance : $E_{learn}^T = (x_1, \dots, x_n, g)$ avec

$$g = \begin{cases} 1 & \text{si } E_{learn} \in A \\ -1 & \text{si } E_{learn} \in B \end{cases}$$

L'algorithme d'apprentissage consiste à donner les vecteurs E_{learn} en entrée du perceptron et comparer la sortie y à la sortie attendue g . Si sortie réelle et sortie attendue sont différents, on va adapter le vecteur de pondération des entrées P_{entree} et le biais b en les incrémentant ou déc rementant d'un pas d'apprentissage α fixée arbitrairement. On répète ces étapes t fois. Plus t est grand, plus le perceptron apprend (Fig 3).

```
initialiser  $P_{entree}$  et  $b$  à 0
initialiser  $\alpha$  ( $0 < \alpha \leq 1$ )
initialiser  $t$ 
pour  $j$  allant de 1 à  $t$  faire
    présenter un vecteur  $E_{learn}$  à l'entrée du perceptron
    calculer la réponse  $y$  du perceptron
    si  $y \neq g$  alors
        pour chaque dimension  $\omega$  de  $P_{entree}$ ,
             $i$  allant de 1 à  $dim(P_{entree})$ , faire
                 $\omega_i \leftarrow \omega_i + x_i * \alpha * g$ 
            fin pour
         $b \leftarrow b + \alpha * g$ 
    fin si
fin pour
```

FIG. 3 – Algorithme du perceptron

2.2 Classement

Après l'apprentissage, les poids et le biais du perceptron sont fixes. On soumet alors des vecteurs E_{entree} et la sortie du perceptron doit correspondre au groupe g dans lequel ils seraient classés (Fig 4).

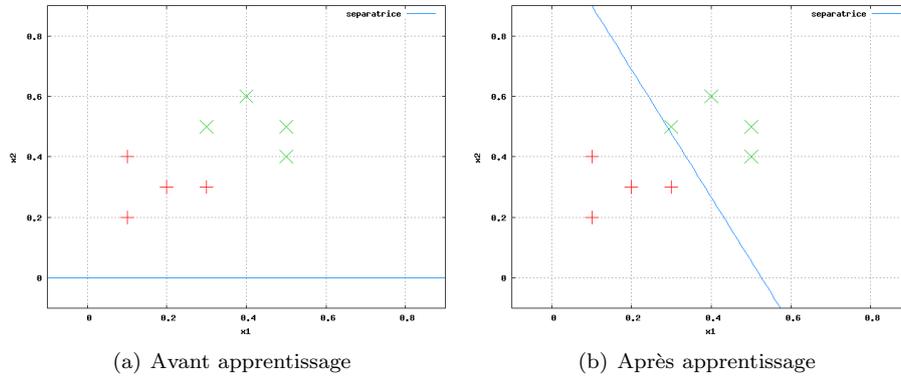


FIG. 4 – Exemple d’apprentissage de séparation entre deux groupes. Ici les vecteurs E_{entree} ont deux dimensions. les vecteurs marqués par un symbole ”*” appartiennent au groupe A, les vecteurs marqués par ”+” sont du groupe B. Ces deux ensembles forment la base d’apprentissage E_{learn} . L’équation de séparatrice résultat du perceptron est dans ce cas $x_1\omega_1 + x_2\omega_2 + b = 0$. Or comme avant l’apprentissage, P_{entree} et b sont à 0, la séparatrice vaut 0. Après apprentissage, poids et biais sont ajustés, on trouve la séparatrice $-0.38x_1 - 0.18x_2 + 0.2 = 0$.

3 Exercice

Imaginons la situation maritime proposée (Fig 5(a)). Nous distinguons deux groupes de bateaux, l’un au sud, l’un au nord. Nous voulons réaliser un programme informatique qui va classer automatiquement tout nouveau bateau arrivant sur la carte dans l’un des deux groupes, soit ”Nord”, soit ”Sud”. Il semble approprié d’utiliser pour cela un perceptron.

Pour localiser les bateaux, nous utiliserons un système de coordonnées dans le plan ($x \rightarrow x_1, y \rightarrow x_2$). Pour des raisons liées au fonctionnement du perceptron, $x_1 \in [0, 1]$ et $x_2 \in [0, 1]$. Le modèle informatique (Fig 5(b)) représente les deux groupes d’apprentissage (”*” et ”+”)

3.1 Apprentissage

Avec Dev-C++, écrivez un programme qui implémente l’algorithme du perceptron. Effectuez un apprentissage à partir des données du cas d’application. En sortie, le programme renverra les poids et biais modifiés.

- $E_{learn}, P_{entree}, b, \alpha$ sont de type float
- utilisez un tableau LEARN [NB.CAS][$dim(E_{learn})$] pour stocker les vecteurs E_{learn} de l’ensemble d’apprentissage
- utilisez un tableau POIDS[$dim(E_{learn}) - 1$] pour stoker le vecteur de poids P_{entree}

Effectuez quelques tests d’apprentissage en faisant varier le pas α et le nombre d’itération d’apprentissage. Représentez les différentes lignes séparatrices résultantes de ces test sur la feuille jointe. Quel serait pour le cas d’application un couple ”efficace” du pas α et du nombre d’itération t ?

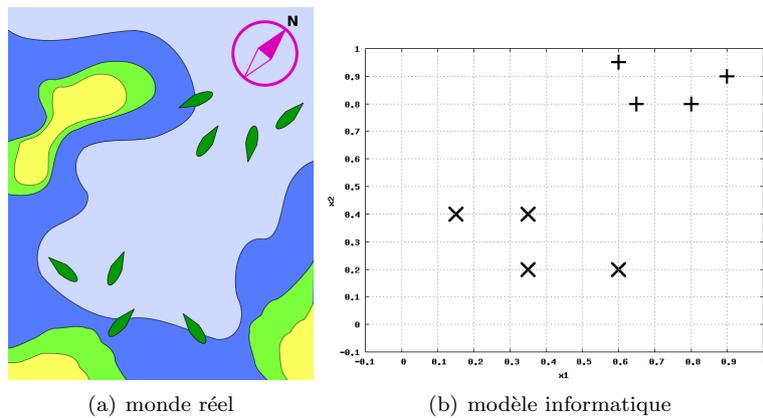


FIG. 5 – Situation d'apprentissage

3.2 test de classification

Après l'apprentissage, demandez la saisie d'un nouveau vecteur de coordonnées correspondant à un bateau que l'on veut classer soit dans le groupe "Nord" soit dans le groupe "Sud". Le programme retournera alors le groupe d'appartenance du vecteur saisi.

- Utilisez la sortie y du perceptron (1 ou -1) pour décider quel groupe retourner

3.3 limites du perceptron

Testez un apprentissage à partir du cas de Fig 6. Dessinez la séparatrice résultante de l'apprentissage sur la feuille jointe. Celle-ci fonctionne-t-elle? Expliquez pourquoi le perceptron n'arrive pas à un résultat d'apprentissage convainquant. En vous documentant sur internet, expliquez quelle serait une méthode pour contourner le problème.

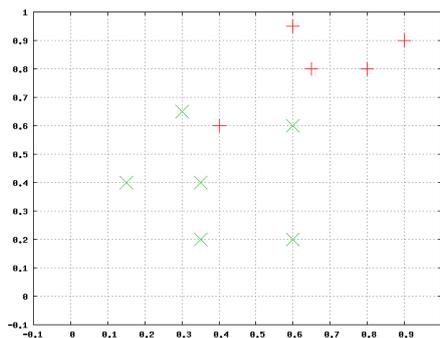


FIG. 6 – Cas d'apprentissage avec deux classes confondues

Expliquez pourquoi le perceptron n'arrive pas à un résultat d'apprentissage convainquant. En vous documentant sur internet, expliquez quelle serait une méthode pour contourner le problème.

