

# Reconnaissance de forme: Les réseaux de neurone

19 février 2010

## Plan

- 1 Introduction
- 2 Perceptron
- 3 Réseau de neurones multi-classes
- 4 Réseau de neurones multi-couches

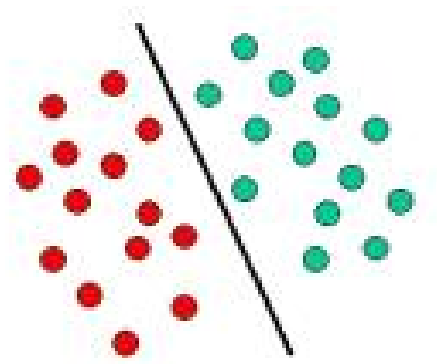
- ① Introduction
- ② Perceptron
- ③ Réseau de neurones multi-classes
- ④ Réseau de neurones multi-couches

On dispose d'une image avec plusieurs objets ayant de nombreux attributs.

On veut classifier ces objets pour les reconnaître.

Objectif : Avoir des algorithmes performants permettant de classifier des données.

Jusque là, algorithmes très simples linéaires à deux classes.



Le problème de classification : un problème d'apprentissage.

On veut apprendre à savoir à l'aide de variables de grande dimension quelle est la classe de l'objet.

De façon générale il existe de très nombreux algorithmes de classification : Arbres, boosting, perceptron et réseaux de neurones multi-couches, SVM, réseaux bayésiens...

Nous allons nous intéresser aux réseaux de neurones...

## 1 Introduction

## 2 Perceptron

Petite histoire des réseaux de neurones...

Concept de perceptron

Apprentissage

## 3 Réseau de neurones multi-classes

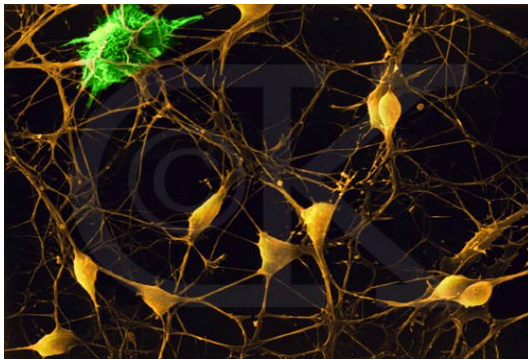
## 4 Réseau de neurones multi-couches

Idée qui s'est développée avec l'apparition de la cybernétique.  
Objectif : créer un système capable d'apprendre. Par exemple des opérateurs logiques.  
A l'image d'un neurone biologique...



## Neurone biologique :

- Les neurones reçoivent des signaux (impulsions électriques) par les dendrites et envoient l'information par les axones.
- Les contacts entre deux neurones (entre axone et dendrite) se font par l'intermédiaire des synapses.
- Les signaux n'opèrent pas de manière linéaire : effet de seuil.



Un statisticien, Franck Rosenblatt, a repris cette idée pour créer le perceptron.

C'est le premier système artificiel capable d'apprendre par expérience à classifier des données y compris quand il y a des erreurs.

Depuis, cet idée d'apprenant artificiel s'est bien développée et a donné lieu à de nombreux algorithmes, notamment les réseaux de neurones, dérivés direct du perceptron, mais aussi aux SVM, boosting...

Concrètement les réseaux de neurones sont moins utilisés qu'avant de par leur complexité et aussi car plusieurs techniques leur sont supérieures.

Mais en reconnaissance de forme cela reste la technique principale.

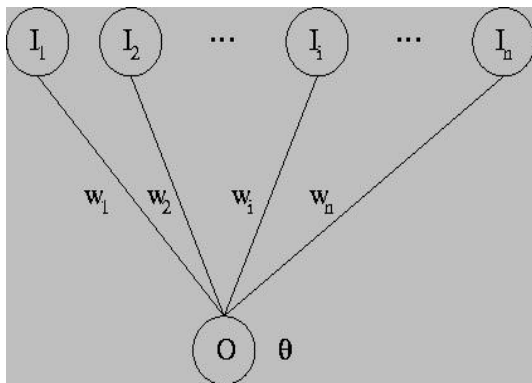


FIG.: Perceptron

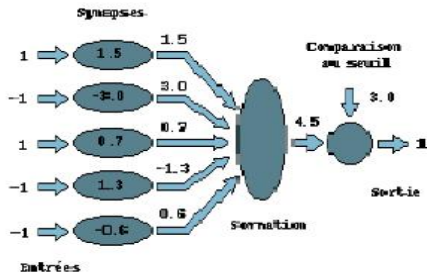


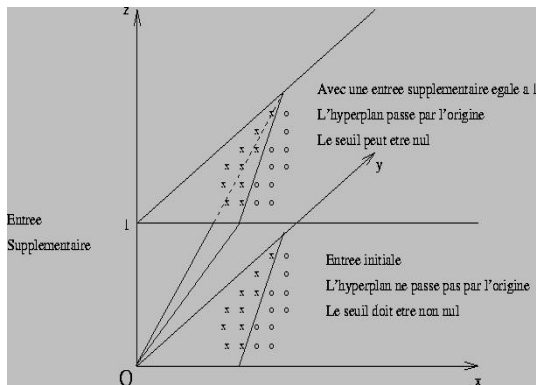
FIG.: Exemple numérique

Ce système est tout simplement un classifieur linéaire. En effet,

$\sum w_i l_i > \theta$  est l'équation d'un hyperplan.

Cela permet de construire deux classes dans l'espace : Par exemple si on a une collection d'objets, on peut classifier entre ronds et carrés.

Remarque : il est facile de rajouter une constante.



**FIG.:** Séparateur linéaire

Exemple : Trouver la droite qui sépare deux ensembles de points



On peut classifier aussi bien des données parfaitement séparables que des données bruitées.

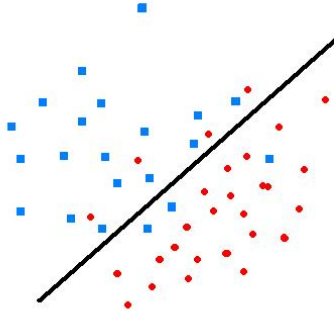


FIG.: Données non linéairement séparables

Exemple : modèle linéaire mais données bruitées

## Remarque sur le bruit :

- Le bruit peut aussi bien être un vrai bruit...
- ... qu'une expression du fait que le modèle linéaire que l'on a choisi n'est pas approprié.

Le modèle du perceptron est posé. Mais comment apprend il sur des données ?

- Déterminer un réseau de neurones : Trouver les coefficients synaptiques.
- On parle de phase d'apprentissage : les caractéristiques du réseau sont modifiées jusqu'à ce que le comportement désiré soit obtenu.
- Echantillon d'apprentissage : exemples représentatifs du comportement ou de la fonction à modéliser. Ces exemples sont sous la forme de couples (entrée , sortie) connus.
- Echantillon de test : pour une entrée quelconque (bruitée ou incomplète), calculer la sortie. On peut alors évaluer la performance du réseau.

## Définitions :

- Apprentissage supervisé : les coefficients synaptiques sont évalués en minimisant l'erreur (entre sortie souhaitée et sortie obtenue) sur un échantillon d'apprentissage.
- Apprentissage non-supervisé : on ne dispose pas d'échantillon d'apprentissage. Les coefficients synaptiques sont déterminés par rapport à des critères de conformité : spécifications générales.
- Sur-apprentissage : on minimise l'erreur sur l'échantillon d'apprentissage à chaque itération mais on augmente l'erreur sur la base d'essai. Le modèle perd sa capacité de généralisation. On apprend le bruit.

Cadre : Apprentissage supervisé

On dispose de données  $S$  avec label (on sait dans quel groupe elles sont).

C'est l'échantillon d'apprentissage.

Exemple :



On va itérer les poids  $w_i$  de notre perceptron jusqu'à ce qu'il sépare correctement notre échantillon d'apprentissage :

$$w_i(t+1) = w_i(t) + \Delta w_i(t)$$

avec  $\Delta w_i(t) = \eta(T - O)l_i$ .

On va ainsi adapter notre séparateur à nos données.

Remarque : rien ne nous dit que nos données sont linéairement séparables...

## 1 Introduction

## 2 Perceptron

## 3 Réseau de neurones multi-classes

Motivation : Plus de classes

Modèle

## 4 Réseau de neurones multi-couches

Imaginons que l'on veuille reconnaître des chiffres : il faut plusieurs classes.

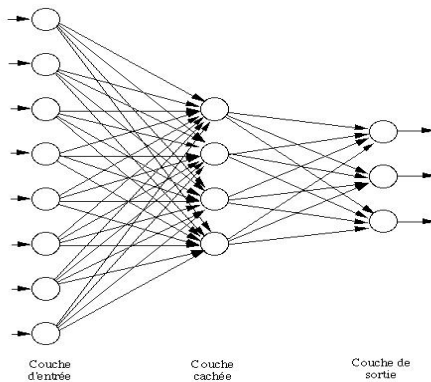


FIG.: Séparateur linéaire

Pour chaque sortie :

$$s_i = \sum w_{i,j} l_j$$

On active la sortie telle que :  $i = \text{Argmax}_i s_i$ .

On peut interpréter  $s_i$  comme la probabilité d'être dans la classe  $i$ .

Exemple de reconnaissance de chiffres :

## ① Introduction

## ② Perceptron

## ③ Réseau de neurones multi-classes

## ④ Réseau de neurones multi-couches

Motivation : au-delà de l'hypothèse linéaire

Modèle



Et si nos séparateurs n'étaient pas linéaires ?

On veut introduire des non linéarités dans notre réseau de neurones pour modéliser une classe plus vaste de problèmes.  
Outil : fonction de lien.

Au lieu d'avoir pour sortie comme dans le perceptron :

$$1_{\sum w_j l_j > \text{seuil}}$$

On veut :

$$f(\text{seuil} + \sum w_j l_j)$$

$f$  est appelée fonction de lien. Cela introduit une non linéarité mais ce n'est pas suffisant.

Idée : Perceptron Multi-couches

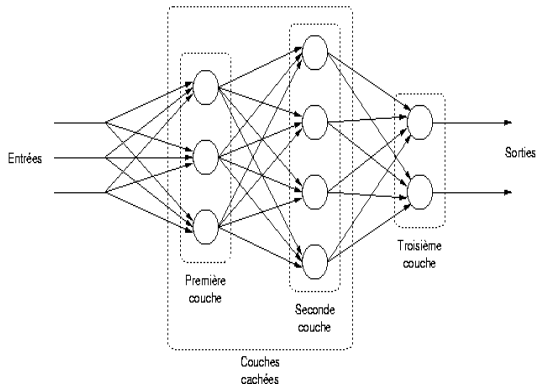


FIG.: Séparateur linéaire