

Cours Apprentissage 2 : Perceptron

Ludovic DENOYER - ludovic.denoyer@lip6.fr

23 mars 2012

Episode précédent

Apprentissage

Au lieu de programmer un ordinateur *manuellement*, donner à l'ordinateur les moyens de se programmer **lui-même**

Pourquoi

- Problèmes trop complexe
- pas d'expert ou expert trop coûteux
- Personnalisation de l'information, grande quantité d'information
- Parce que c'est rigolo....

Episode précédent

Apprentissage

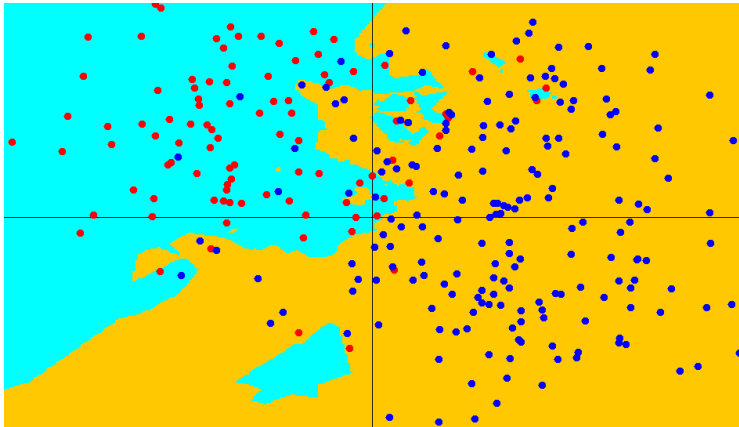
Au lieu de programmer un ordinateur *manuellement*, donner à l'ordinateur les moyens de se programmer **lui-même**

Apprentissage supervisé

- On se limite à la problématique de classification binaire
- Fournir au système un **ensemble d'exemples étiquetés d'apprentissage**
- $f_{\theta} : \mathbb{R}^N \rightarrow [-1; +1]$

Se programmer **lui-même** = trouver les **paramètres optimaux**

Exemple : KNN



Paramètres

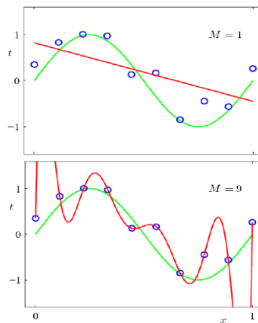
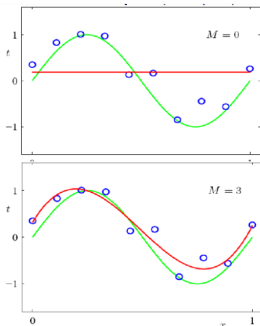
- f_{θ} = k-plus-proches voisins
- θ = ensemble des exemples d'apprentissage

Généralisation

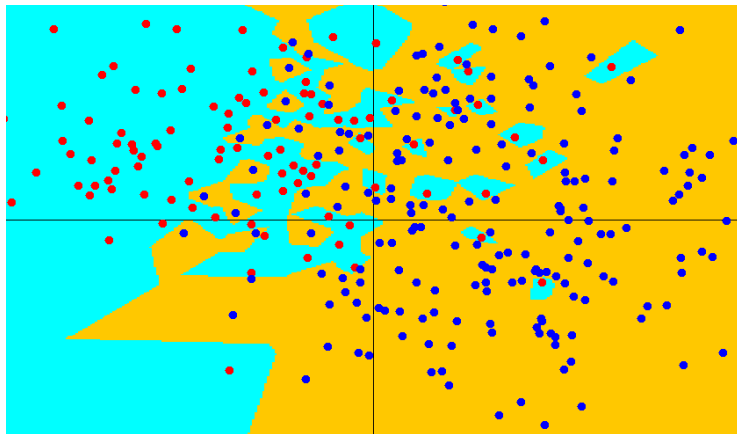
Généralisation

Un bon modèle d'apprentissage est un modèle qui **généralise bien** :

- Evaluation du modèle sur **une base de test**
- Compromis entre un modèle **compliqué** et un modèle **simple**

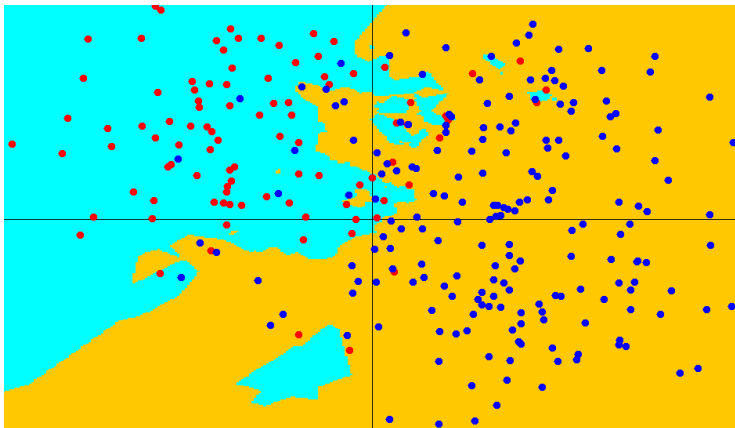


Généralisation



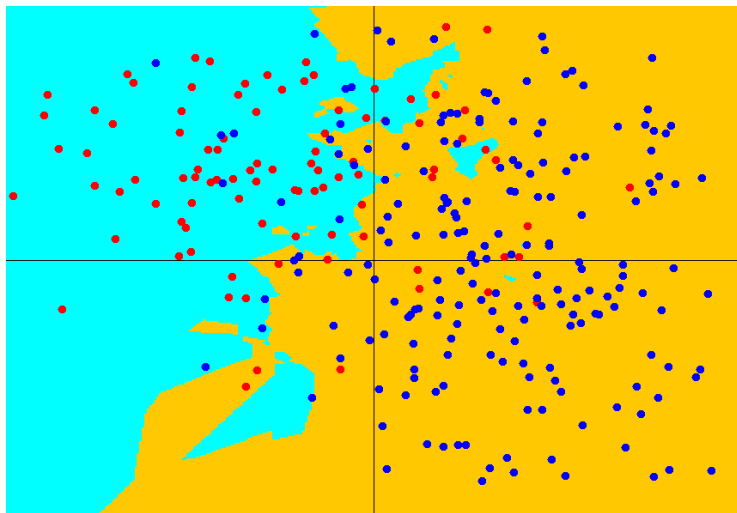
$$K = 1$$

Généralisation

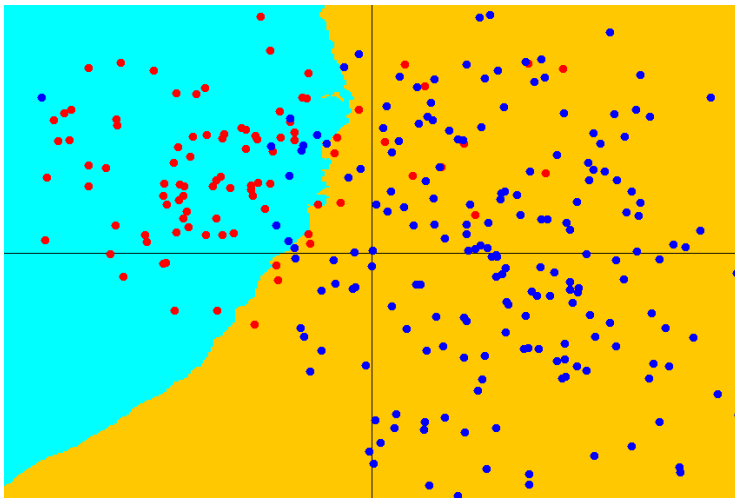


$$K = 3$$

Généralisation

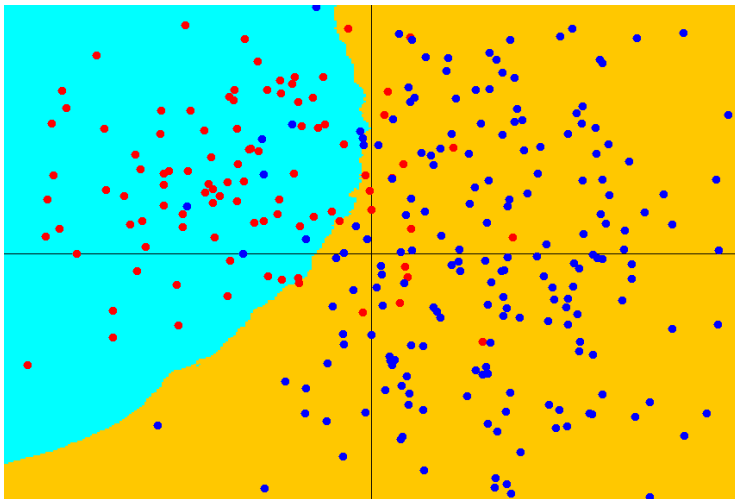
 $K = 5$

Généralisation



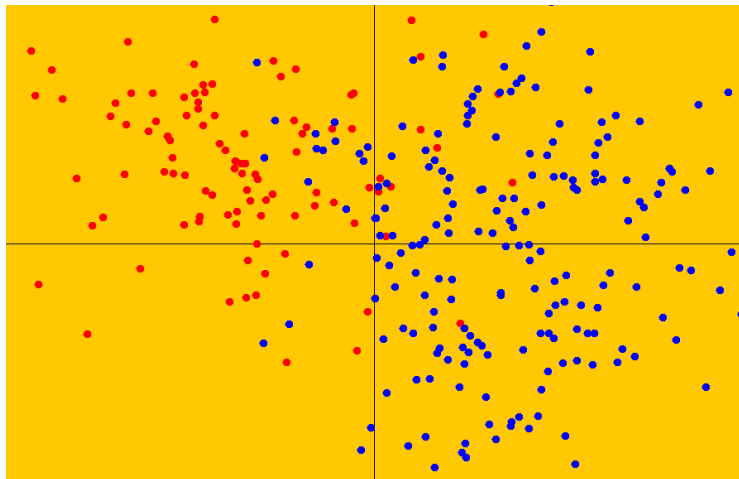
$K = 20$

Généralisation



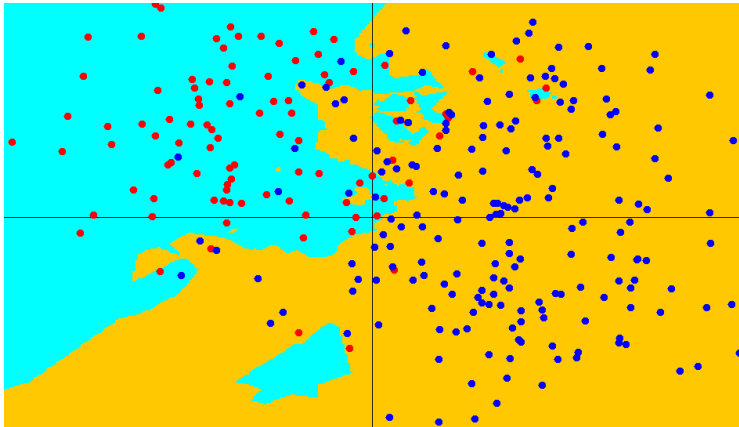
$K = 100$

Généralisation



$$K = 200$$

Exemple : KNN



Paramètres

- f_{θ} = k-plus-proches voisins
- θ = ensemble des exemples d'apprentissage

Problématique du jour

Constat

1-NN est un modèle :

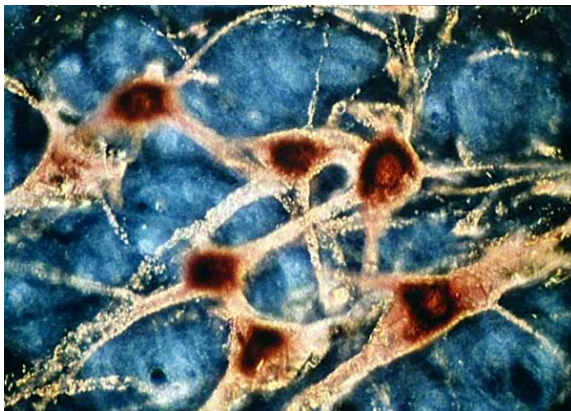
- Trop complexe
- Qui possède beaucoup trop de paramètres

Problématique

Quel modèle (de classification) utiliser qui soit :

- Simple
- Peu de paramètres
- Efficace

Inspiration Biologique



Neurone

- Machine simple (en apparence)
- qui permet d'apprendre

Inspiration biologique

- Le cerveau naturel : un modèle très séduisant
 - Robuste et tolérant aux fautes
 - Flexible. Facilement adaptable
 - S'accommode d'informations incomplètes, incertaines, vagues, bruitées...
 - Massivement parallèle
 - Capable d'apprentissage

Inspiration biologique

- Le cerveau naturel : un modèle très séduisant
 - Robuste et tolérant aux fautes
 - Flexible. Facilement adaptable
 - S'accommode d'informations incomplètes, incertaines, vagues, bruitées...
 - Massivement parallèle
 - Capable d'apprentissage
- Neurones
 - 10^{11} neurones dans le cerveau humain
 - 10^4 connexions (synapses + axones) / neurone
 - Potentiel d'action / période réfractaire / neuro-transmetteurs
 - Signaux excitateurs / inhibiteurs

Les attraits pratiques

- Neurones
- Calculs parallélisables
- Implantables directement sur circuits dédiés
- Robustes et tolérants aux fautes (calculs et représentations distribués)
- Algorithmes simples
- D'emploi très général

Les défauts

- Opacité des raisonnements
- Opacité des résultats

Historique

- Prémisses

- Mc Culloch & Pitts (1943) : 1er modèle de neurone formel.
Rapport neurone et calcul logique : base de l'intelligence artificielle.
- Règle de Hebb (1949) : apprentissage par renforcement du couplage synaptique

Historique

- Prémisses
 - Mc Culloch & Pitts (1943) : 1er modèle de neurone formel. Rapport neurone et calcul logique : base de l'intelligence artificielle.
 - Règle de Hebb (1949) : apprentissage par renforcement du couplage synaptique
- Premières réalisations
 - Adaline (Widrow-Hoff, 1960)
 - Perceptron (Rosenblatt, 1958-1962)
 - Analyse de Minsky & Papert (1969)

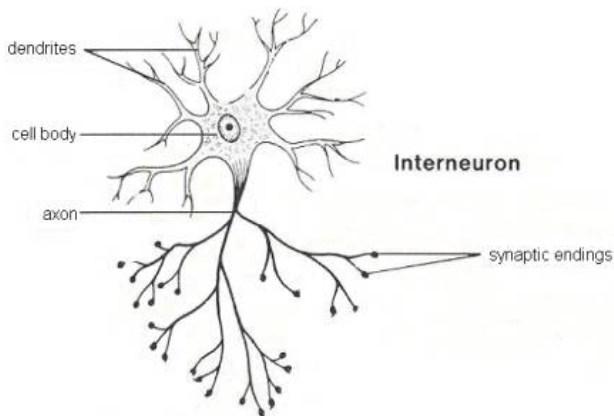
Historique

- Prémisses
 - Mc Culloch & Pitts (1943) : 1er modèle de neurone formel. Rapport neurone et calcul logique : base de l'intelligence artificielle.
 - Règle de Hebb (1949) : apprentissage par renforcement du couplage synaptique
- Premières réalisations
 - Adaline (Widrow-Hoff, 1960)
 - Perceptron (Rosenblatt, 1958-1962)
 - Analyse de Minsky & Papert (1969)
- Nouveaux modèles
 - Kohonen (apprentissage compétitif), ...
 - Hopfield (1982) (réseau bouclé)
 - Perceptron Multi-Couches (1985)

Historique

- Prémisses
 - Mc Culloch & Pitts (1943) : 1er modèle de neurone formel. Rapport neurone et calcul logique : base de l'intelligence artificielle.
 - Règle de Hebb (1949) : apprentissage par renforcement du couplage synaptique
- Premières réalisations
 - Adaline (Widrow-Hoff, 1960)
 - Perceptron (Rosenblatt, 1958-1962)
 - Analyse de Minsky & Papert (1969)
- Nouveaux modèles
 - Kohonen (apprentissage compétitif), ...
 - Hopfield (1982) (réseau bouclé)
 - Perceptron Multi-Couches (1985)
- Analyse et développements
 - Théorie du contrôle, de la généralisation (Vapnik), ...

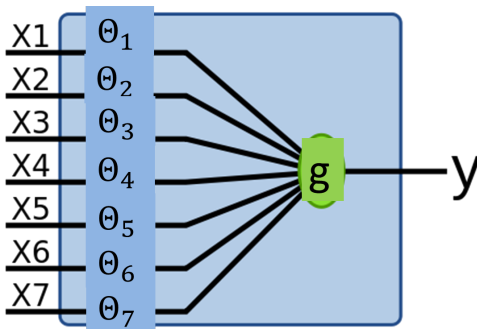
Inspiration Biologique



Neurone

- Machine simple (en apparence)
- qui permet d'apprendre

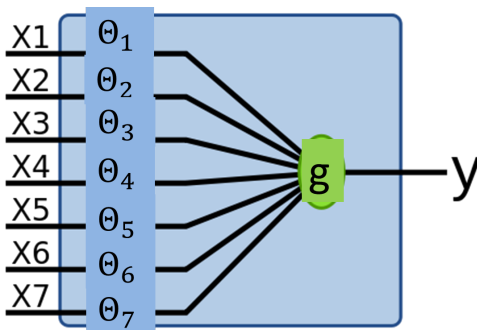
Perceptron de Rosenblatt (1960)



Perceptron de Rosenblatt

C'est le premier système artificiel capable d'apprendre par expérience, y compris lorsque son instructeur commet quelques erreurs (ce en quoi il diffère nettement d'un système d'apprentissage logique formel).

Percetron de Rosenblatt (1960)

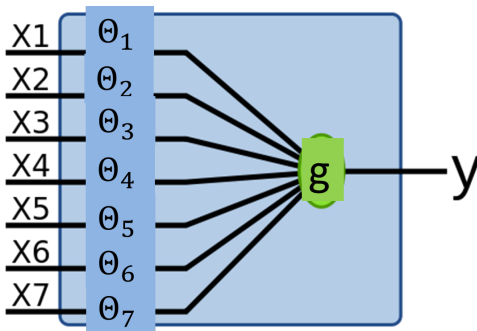


Fonction de décision

$$f_{\theta}(x) = g(x_1 * \theta_1 + x_2 * \theta_2 + x_3 * \theta_3 + x_4 * \theta_4 + x_5 * \theta_5 + x_6 * \theta_6 + x_7 * \theta_7)$$

Décision : $f_{\theta}(x) > 0$?

Percetron de Rosenblatt (1960)



Fonction de décision

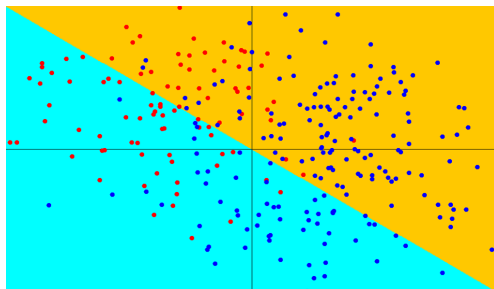
$$f_{\theta}(x) = g\left(\sum_{i=1}^N x_i * \theta_i\right)$$

$$\text{Si } g(x) = x : f_{\theta}(x) = \langle x; \theta \rangle$$

Percetron de Rosenblatt (1960)

Fonction de décision

$$f_{\theta}(x) = \langle x; \theta \rangle$$

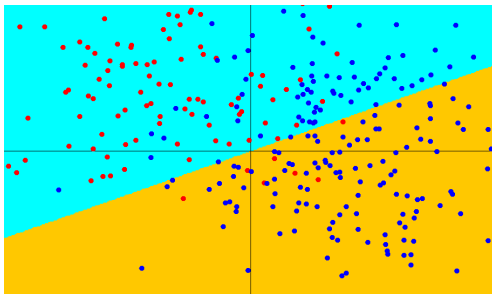


$$\theta = (1; 1)$$

Percetron de Rosenblatt (1960)

Fonction de décision

$$f_{\theta}(x) = \langle x; \theta \rangle$$



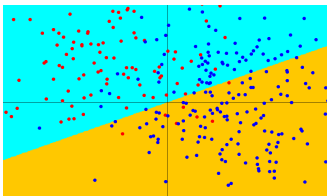
$$\theta = (0.3; -0.5)$$

Percetron de Rosenblatt (1960)

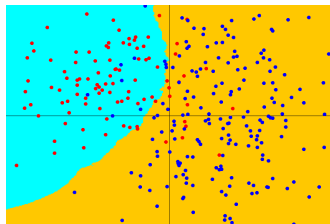
Fonction de décision

Le perceptron est une **machine de classification linéaire**. Il modélise un hyperplan dans un espace vectoriel ^a

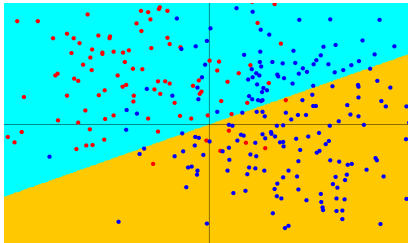
^aUn truc tout droit qui coupe en deux



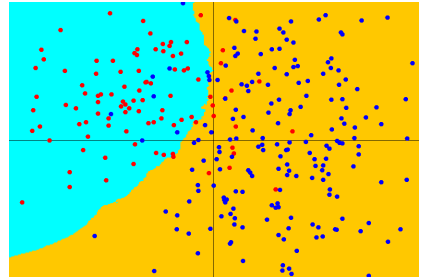
Perceptron



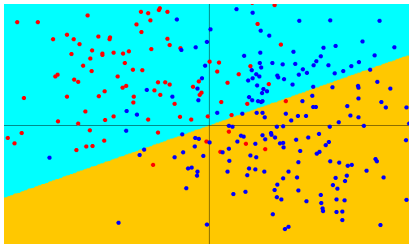
KNN



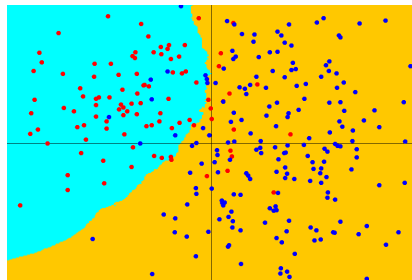
2 paramètres



600 paramètres



Temps de calcul très courts



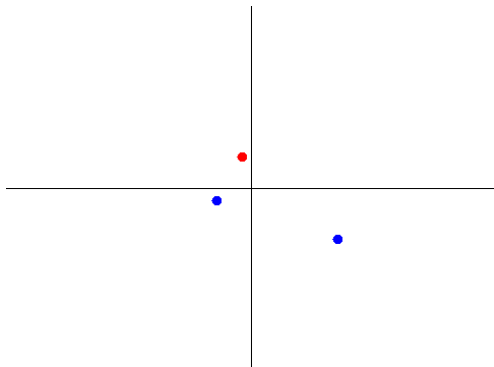
Temps de calcul prohibitif

Problème

Le perceptron modélise une **frontière linéaire**..... Est-ce que ça peut apprendre quelque chose ?

Problème

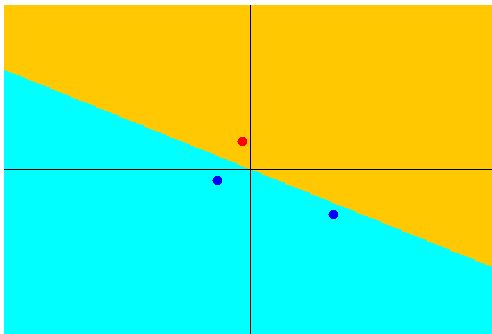
Le perceptron modélise une **frontière linéaire**..... Est-ce que ça peut apprendre quelque chose ?



Linéairement séparable

Problème

Le perceptron modélise une **frontière linéaire**..... Est-ce que ça peut apprendre quelque chose ?



Informations....

3 points en dimension 2 sont toujours **linéairement séparables**

Linéairement séparable

Informations....

3 points en dimension 2 sont toujours **linéairement séparables**

4 points en dimension 3 sont toujours **linéairement séparables**

...

n points en dimension $n - 1$ sont toujours **linéairement séparables**

Le perceptron est un modèle adapté à l'apprentissage **dans les espaces de grande dimension.**

- texte
- image
- séries financières
- ...

L'algorithme du perceptron

Algorithme d'apprentissage

- Initialiser θ aléatoirement
 - Répéter :
 - Pour $i = 1$ à N
 - Si $y^i * (< \theta; x^i >) \leq 0$ alors $\theta = \theta + \epsilon.y^i.x^i$
 - Jusqu'à convergence
-
- Algorithme de correction d'erreur
 - ϵ peut être fixe ou variable (décroissant)
 - Démonstration

L'algorithme du perceptron

Algorithme d'apprentissage

- Initialiser θ aléatoirement
- Répéter :
 - Pour $i = 1$ à N
 - Si $y^i * (< \theta; x^i >) \leq 0$ alors $\theta = \theta + \epsilon.y^i.x^i$
- Jusqu'à convergence

DEMONSTRATION!!!

Théorème de convergence du perceptron - Novikov 1962

- Si :
 - $\exists R/\forall x, ||x|| \leq R$
 - les données peuvent être séparées avec une marge ρ

$$\sup_{\theta} \min_i y^i \cdot \langle x^i; \theta \rangle \geq \rho \quad (1)$$

- L'ensemble d'apprentissage peut être présenté un nombre suffisant de fois
- Alors :

L'algorithme converge après au plus $\frac{R^2}{\rho^2}$ corrections

Propriétés sur l'erreur de généralisation

- Si :
 - Les données sont séparables
 - Elles sont en nombre infini
 - : après la k ème correction, les $m_k = \frac{1+2 \ln k - \ln \eta}{-\ln(1-\epsilon)}$ données présentées sont reconnues correctement
- Alors :

- Le perceptron converge en $l \leq \frac{1+4 \ln \frac{R}{\rho} - \ln \eta}{-\ln(1-\epsilon)} \cdot \frac{R^2}{\rho^2}$ étapes
- avec une probabilité $1 - \eta$, l'erreur de test est $\leq \epsilon$

Perceptron : résumé

Perceptron

- Inventé en 1960 - premier modèle d'apprentissage...
- ...dont on a une preuve de convergence (il apprend quelque chose).
- Permet de faire de la classification **linéaire**...
- ... ce qui est bien suffisant en grande dimension.
- Algorithme itératif d'apprentissage...
- ...ne corrige que si il y a une erreur.

Réseau de neurones

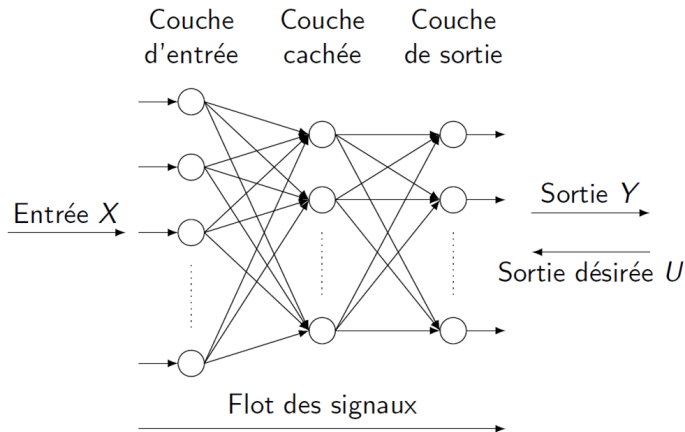
Idée...

Si on en mettait plusieurs bout à bout ?

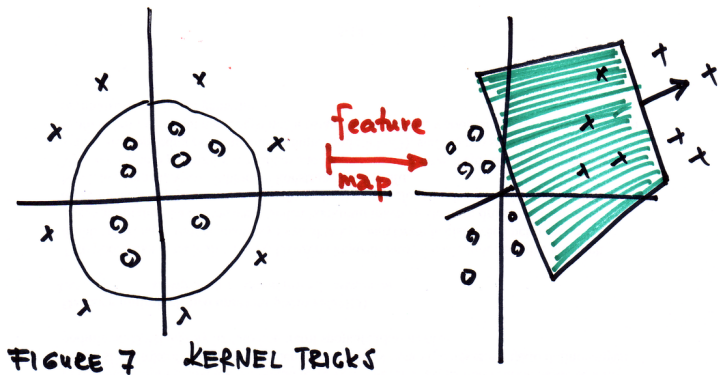
Réseau de neurones

- Apparition en 1986
- Constitué de plusieurs couches de neurones

Réseau de neurones



Réseau de neurone



Conclusion

Le perceptron :

- Classification linéaire
- Modélisation par un hyper-plan
- Algorithme d'apprentissage itératif

TD/TP :

- Implémentation et comparaison avec KNN