Perceptron
Réseau de neurones (I)

Cours 4

Nicolas Baskiotis
nicolas.baskiotis@lip6.fr

Master 1 DAC

équipe MLIA, Laboratoire d'Informatique de Paris 6 (LIP6)
Université Pierre et Marie Curie (UPMC)

S2 (2014-2015)

Plan

Résumé des épisodes

Notions abordées

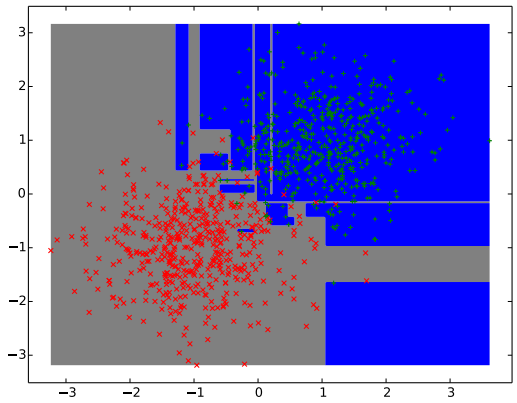
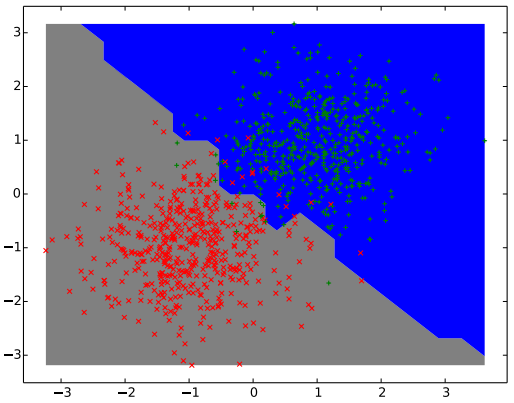
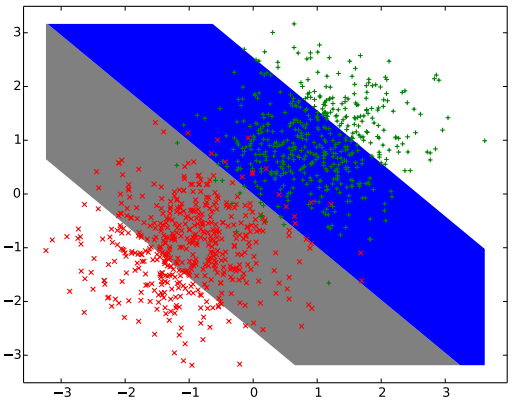
- Ensemble d'apprentissage : $\mathcal{D} = \{(\mathbf{x^i}, y^i) \in \mathcal{X} \times \mathcal{Y}\}_{i=1,\ldots,N}, \mathcal{X} \in \mathbb{R}^d$
- Ensemble de test
- Régression : $\mathcal{Y} \in \mathbb{R}$
- Apprentissage supervisée : $\mathcal{Y}$ discret
- Apprentissage non supervisée : pas de $\mathcal{Y}$
- Apprentissage : trouver $f : \mathcal{X} \to \mathcal{Y}$ qui fait le moins d'erreurs
- Erreur : $0-1$, moindres carrés
- Sélection de modèle : validation croisée

Approches et algorithmes

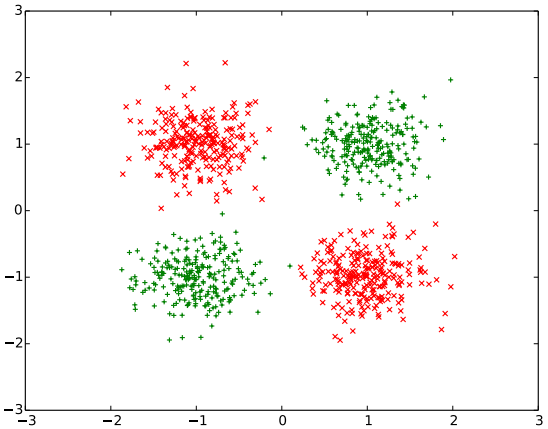
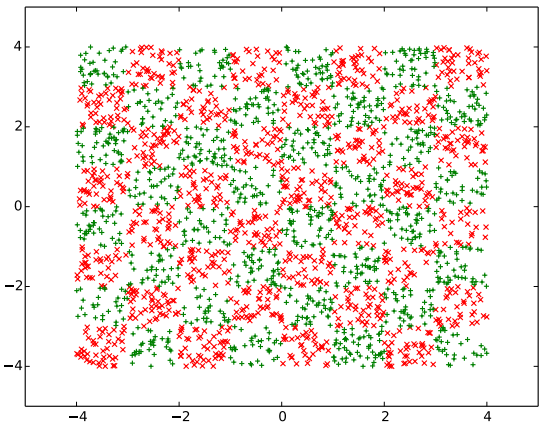
- Estimation de densité
- Arbres de décision
- Classifieur bayésien, décision bayésienne, vraissemblance
- Régression logistique
- Descente de gradient

On réfléchit un peu

Quelle approche pour ce cas ?

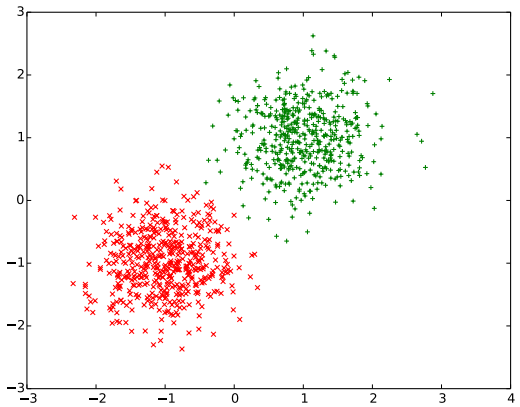




Et pour ces cas ?

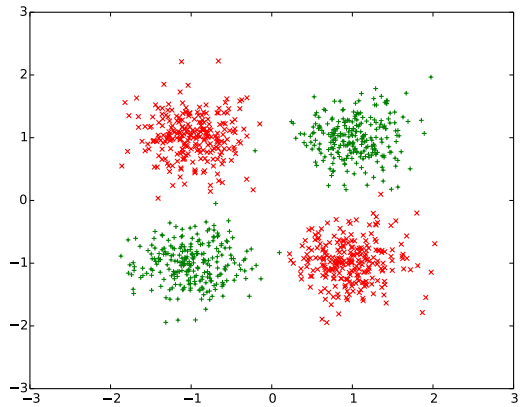



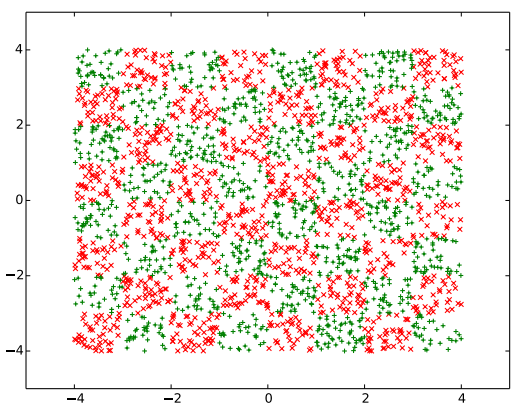
Espace linéairement séparable

Linéairement séparable



Non linéairement séparable





Conclusion (temporaire)

Importance :

- de l'espace de fonctions considérées (choisie a priori)
- du paramètrage des algorithmes

$\rightarrow$ notion d'expressivité . . . *à suivre.*

Plan

Inspiration biologique

Le cerveau

- Robuste, tolérant aux fautes
- Flexible, sait s'adapter
- Gère les informations incomplètes
- Capable d'apprendre

Composé de neurones !

- 10^{11} neurones dans un cerveau humain
- 10^{4} connexions par neurones
- Potentiel d'action, neuro-transmetteurs, période réfractaire
- Signaux excitateurs / inhibiteurs

Problèmes

- Opacité des raisonnements
- Opacité des résultats

Inspiration biologique

Le cerveau

- Robuste, tolérant aux fautes
- Flexible, sait s'adapter
- Gère les informations incomplètes
- Capable d'apprendre

Composé de neurones !

- 10^{11} neurones dans un cerveau humain
- 10^{4} connexions par neurones
- Potentiel d'action, neuro-transmetteurs, période réfractaire
- Signaux excitateurs / inhibiteurs

Problèmes

- Opacité des raisonnements
- Opacité des résultats

Inspiration biologique

Le cerveau

- Robuste, tolérant aux fautes
- Flexible, sait s'adapter
- Gère les informations incomplètes
- Capable d'apprendre

Composé de neurones !

- 10^{11} neurones dans un cerveau humain
- 10^{4} connexions par neurones
- Potentiel d'action, neuro-transmetteurs, période réfractaire
- Signaux excitateurs / inhibiteurs

Problèmes

- Opacité des raisonnements
- Opacité des résultats

Historique

Prémisses

- Mc Cullch et Pitts (1943) : 1er modèle de neurone formel. Base de l'IA
- Règle de Hebb (1949) : apprentissage par renforcement du couplage synaptique

Premières réalisations

- Adaline (Widrow-Hoff, 1960)
- Perceptron (Rosenblatt, 1958-1962)
- Analyse de Minsky et Papert (1969)

Développement

- Réseau bouclé (Hopfield 1982)
- Réseau multi-couches (1985)

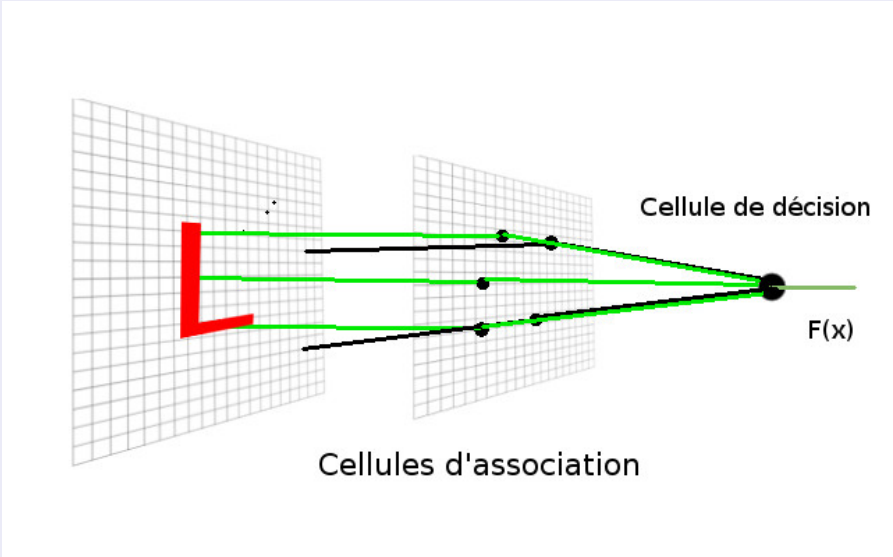
Deuxième renaissance

- Réseaux profonds (2000-)

Le perceptron de Rosenblatt (1960)

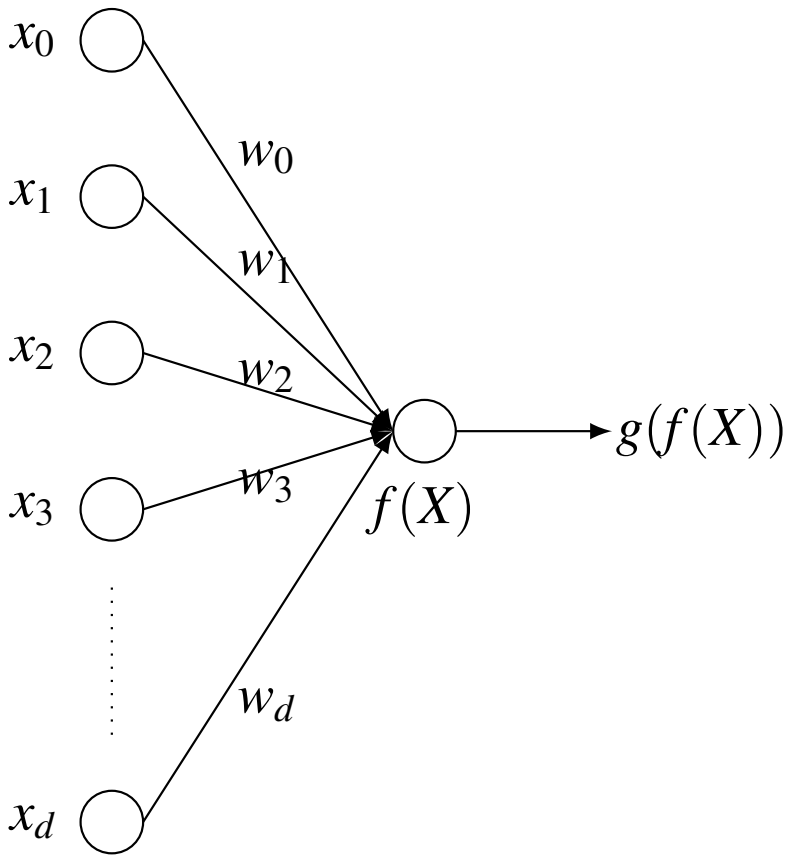
L'idée

- Reconaissance de forme (*pattern*) entre deux classes
- Inspirée cortex visuel



- Chaque cellule d'association produit une sortie $f_i(S)$ en fonction d'un stimulus
- La cellule de décision répond selon une fonction seuil $f_d(\sum w_i f_i(S_i))$

Formalisation



Le perceptron considère

- $f(\mathbf{x}) =< \mathbf{x}, \mathbf{w} >= \sum_{i=1}^{d} x_i w_i$
- Fonction de décision : $g(x) = sign(x)$

$\rightarrow$ Sortie : $g(f(\mathbf{x})) = sign(< \mathbf{x}, \mathbf{w} >)$

Considérations géométriques

Soit $y(x)$ la sortie attendue :

- Que représente w par rapport à la séparatrice ?
- Que représente $< w\mathbf{x} >$?
- Que représente $y(x) < w\mathbf{x} >$?
- A quoi correspond la règle de mise à jour :
  - Si $(y(x) < w.\mathbf{x} >) > 0$ ne rien faire
  - Si $(y(x) < w.\mathbf{x} >) < 0$ corriger $w = w + y(x)x$?

Algorithme de résolution

Algorithme du perceptron

- Initialiser au hasard w
- Tant qu'il n'y a pas convergence :
  - pour tous les exemples (x^i, y^i) :
    - si $(y^i < w.\mathbf{x}^i >) < 0$ alors $w = w + \epsilon y^i x^i$
- Décision : $f(x) = sign(< w\mathbf{x} >)$

Théorème de convergence (Novikov, 1962)

- Si
  - $\exists R, \forall x : \|x\| \leq R$
  - les données peuvent être séparées avec une marge ρ
  - l'ensemble d'apprentissage est présenté au perceptron un nombre suffisant de fois
- alors après au plus R^2/ρ^2 corrections, l'algorithme converge.

Autre formulation : descente de gradient !

Hinge loss

$l(f(x), y) = max(0, \alpha - yf(x))$

Dans le cas du perceptron

- $l(f_w(x), y) = max(0, -y < w.\mathbf{x} >)$
- $\nabla_w l(f_w(x), y) = \begin{cases} 0 & \text{si } (y < w.\mathbf{x} >) > 0 \\ -yx_i & \text{sinon} \end{cases}$

Pourquoi ce changement dans la fonction de coût ?

Autre formulation : descente de gradient !

Hinge loss

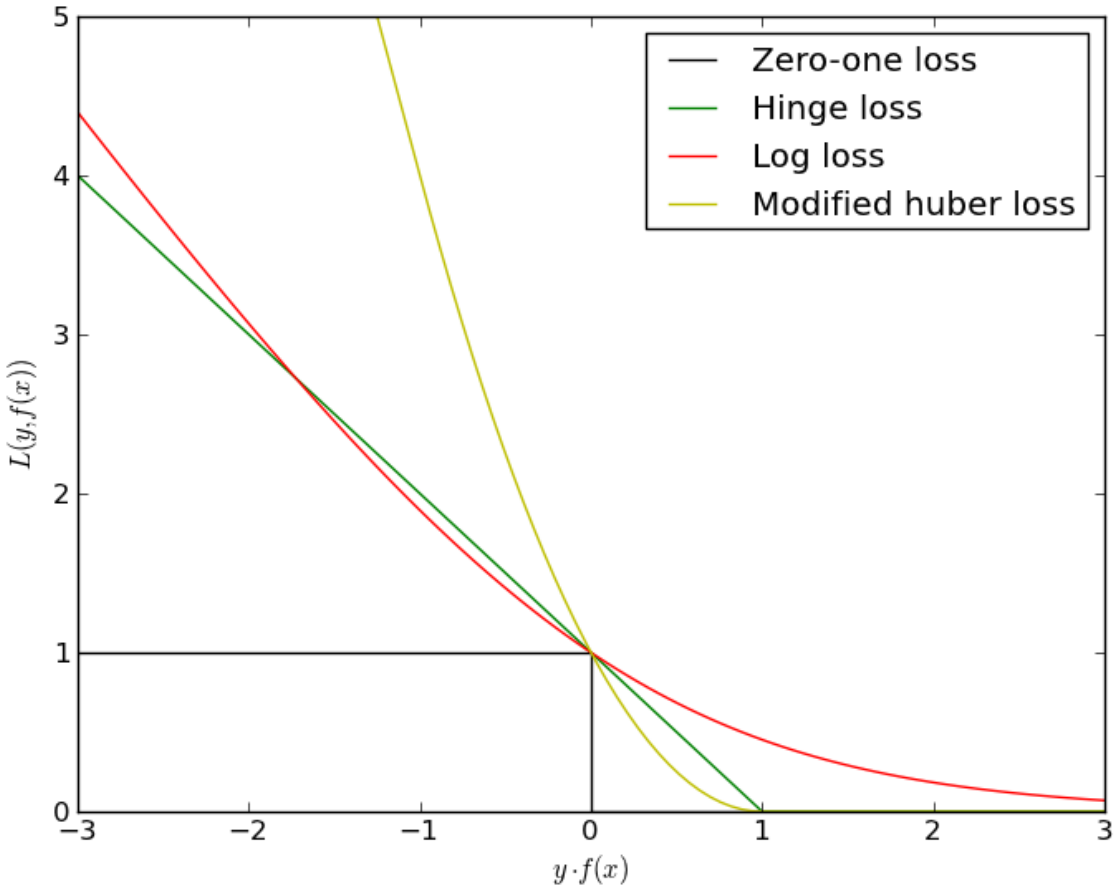
$l(f(x), y) = max(0, \alpha - yf(x))$

Dans le cas du perceptron

- $l(f_w(x), y) = max(0, -y < w.\mathbf{x} >)$
- $\nabla_w l(f_w(x), y) = \begin{cases} 0 & \text{si } (y < w.\mathbf{x} >) > 0 \\ -yx_i & \text{sinon} \end{cases}$

Pourquoi ce changement dans la fonction de coût ?

Et pourquoi pas d'autres erreurs ?



Variantes de l'algorithme

- Cas hors-ligne (ou batch) :
  Pour chaque époque (correction de w), on itère sur toute la base d'exemples
- Cas en-ligne (stochastique) :
  Une correction de w est faîte par rapport à un exemple tiré au hasard dans la base.

Avantages et inconvénients ?

- Batch : plus stable, plus rapide
- Stochastique : bien meilleur tolérance au bruit !

Variantes de l'algorithme

- Cas hors-ligne (ou batch) :
  Pour chaque époque (correction de w), on itère sur toute la base d'exemples
- Cas en-ligne (stochastique) :
  Une correction de w est faîte par rapport à un exemple tiré au hasard dans la base.

Avantages et inconvénients ?

- Batch : plus stable, plus rapide
- Stochastique : bien meilleur tolérance au bruit !