

Quelques notes sur l'apprentissage machine

felipe@iro.umontreal.ca

RALI
Dept. Informatique et Recherche Opérationnelle
Université de Montréal



V0.00001

Last compiled: 22 février 2018



Plan

Perceptron moyenné

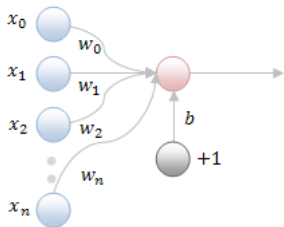
Perceptron structuré

Plan

Perceptron moyenné

Perceptron structuré

Perceptron (Rosenblatt, 1957)



- ▶ x_j input (valeur réelle)
- ▶ w_j poids synaptique (valeur réelle)
- ▶ b biais (peut être représenté par un poids supplémentaire et une entrée fixe)
- ▶ $h = \sum_j x_j \times w_j + b$ est le signal arrivant au neurone
- ▶ l'activation du neurone est déterminée par une fonction d'activation $\hat{y} \equiv f(h)$ (ex : $f = \text{sign}$)

Perceptron (Rosenblatt, 1957)

- ▶ soit $(x_i, y_i)_{i \in [1, L]}$ un corpus d'entraînement où y_i (ici à valeur dans $\{-1, +1\}$) est la réponse (supervision) et $x_i \in \mathbb{R}^{n+1}$
- ▶ apprendre les w_j peut se faire **en ligne** en ajustant itérativement les poids une fois chaque observation x_i rencontrée :
$$\forall j \in [0, n], w_j \leftarrow w_j + \eta(y_i - \hat{y}_i) \cdot x_i$$

où η est le **learning rate**

résout des problèmes linéairement séparables



Voted-Perceptron [*Freund and Schapire, 1999*]

- ▶ Entraînement en ligne :

Require: $\{(\mathbf{x}_i, y_i)\}_{i \in [1, L]}$ où $y_i \in \{+1, -1\}$ et $x_i \in \mathbb{R}^n$

Ensure: a pool of K perceptrons $\{(\mathbf{v}_k, c_k)\}_{k \in [1, K]}$, $\mathbf{v}_k \in \mathbb{R}^n$, $c_k \in \mathbb{N}$

$k, c_1 \leftarrow 0$

$\mathbf{v}_1 \leftarrow [0 \dots 0]^T$

for all epoch **do**

for all $i \in [1, L]$ **do**

$\hat{y} \leftarrow \text{sign}(\mathbf{v}_k \cdot \mathbf{x}_i)$

if $\hat{y} \neq y_i$ **then**

$\mathbf{v}_{k+1} \leftarrow \mathbf{v}_k + y_i \mathbf{x}_i$

$c_{k+1} \leftarrow 1$

$k \leftarrow k + 1$

else

$c_k \leftarrow c_k + 1$

Garanties de convergence (même dans les cas non linéairement séparables)

Voted-perceptron [*Freund and Schapire, 1999*]

- ▶ Test :

$$\hat{y} = \text{sign} \left(\sum_k c_k \cdot \text{sign}(\mathbf{v}_k \cdot \mathbf{x}) \right)$$

- ▶ requiert la sauvegarde de K perceptrons
- ▶ calcul en $O(K \times n)$. On peut également ne retenir qu'un sous-ensemble des perceptrons (à l'extrême un seul, par exemple celui de plus fort c_k).



Plan

Perceptron moyenné

Perceptron structuré

Perceptron structuré [Collins, 2002]

```
w0, wa ← 0  
repeat  
  for all example (x, y) ∈ D do  
     $\hat{y} = \operatorname{argmax}_y \mathbf{w}_0^T \Phi(x, y)$   
    if  $\hat{y} \neq y$  then  
       $\mathbf{w}_0 \leftarrow \mathbf{w}_0 + \Phi(x, y) - \Phi(x, \hat{y})$   
       $\mathbf{w}_a \leftarrow \mathbf{w}_a + c\Phi(x, y) - c\Phi(x, \hat{y})$   
       $c \leftarrow c + 1$   
until converged  
return  $\mathbf{w}_0 - \mathbf{w}_a/c$ 
```



Collins, M. (2002).

Discriminative training methods for hidden markov models : Theory and experiments with perceptron algorithms.

In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 1–8.



Freund, Y. and Schapire, R. E. (1999).

Large margin classification using the perceptron algorithm.

Mach. Learn., 37(3) :277–296.