



Chapitre 4: Perceptron simple

Ibrahima Sy

Ph.d Cand. in Machine Learning

Université Cheikh Anta Diop de Dakar(UCAD)
Master Modélisation Statistique et Informatique(MSI)
Faculté des Sciences et Techniques (FST)

May 5, 2021

Plan

Présentation

Fonctionnement de L'algorithme

Règle d'apprentissage

Convergence et séparabilité

Courbe d'apprentissage

References

Présentation

Le **perceptron simple** est un algorithme d'apprentissage supervisé de classifieurs binaires (c'est-à-dire séparant deux classes). Il a été inventé en 1957 par *Frank Rosenblatt* au laboratoire d'aéronautique de l'université Cornell. Il s'agit d'un neurone formel muni d'une règle d'apprentissage qui permet de déterminer automatiquement les poids synaptiques de manière à séparer un problème d'apprentissage supervisé. Si le problème est linéairement séparable, un théorème assure que la règle du perceptron permet de trouver une séparatrice entre les deux classes.

Perceptron Simple

1. base des réseaux de neurones
2. Entré : Vecteurs $x \in \mathbb{R}^m$
3. Neurone de sortie : $\hat{y} \in \mathbb{R}$
4. Transformation affine : $s = \omega^\top x + b$
5. Transformation non linéaire (fonction activation) : $f_w(s) = \hat{y}$

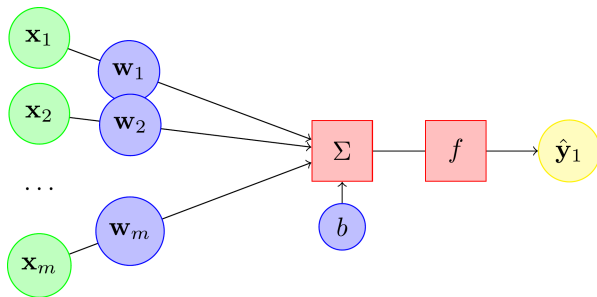
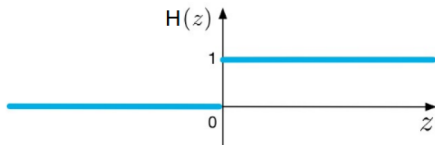


Figure 1: Neurone formel

Fonction Activation

- fonction Heaveside

$$f_w(z) = \text{Threshold}(z) = \begin{cases} 1 & z \geq 0 \\ 0 & \text{sinon} \end{cases}$$

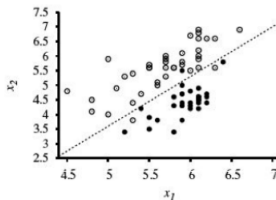
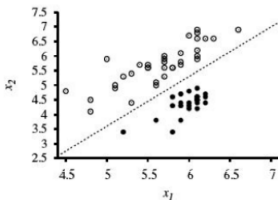


Règle d'apprentissage

- ▶ L'algorithme d'apprentissage doit adapter la valeur des paramètres (c'est-à-dire les poids et le biais) de façon à ce que $f_w(s)$ soit la bonne réponse sur les données d'entraînement
- ▶ **Algorithme du perceptron:**
 1. pour chaque paire $(\mathbf{x}_t, y_t) \in D$
 - a. calculer $h_w(\mathbf{x}_t) = \text{Threshold}(\mathbf{w} \cdot \mathbf{x}_t)$
 - b. si $y_t \neq h_w(\mathbf{x}_t)$
 - $\mathbf{w} \leftarrow \mathbf{w} + \alpha(y_t - h_w(\mathbf{x}_t))\mathbf{x}_t$ (mise à jour des poids et biais)
 2. retourner à 1 jusqu'à l'atteinte d'un critère d'arrêt (nb. maximal d'itérations atteint ou nb. d'erreurs est 0)
- ▶ La mise à jour des poids est appelée la règle **d'apprentissage du Perceptron**. Le multiplicateur α est appelé le **taux d'apprentissage**

Convergence et séparabilité

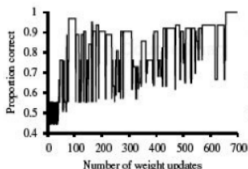
- ▶ Si les exemples d'entraînement sont **linéairement séparables** (gauche), l'algorithme est garanti de converger **une solution avec une erreur nulle** sur l'ensemble d'entraînement, quel que soit le choix de α



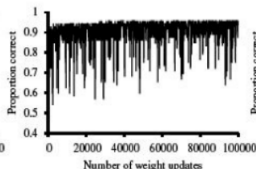
- ▶ Si **non-séparable linéairement** (droite), pour garantir la convergence à une solution avec la plus petite erreur possible en entraînement, on doit décroître le taux d'apprentissage, par ex. selon $\alpha_k = \frac{\alpha}{1+\beta_k}$

Courbe d'apprentissage

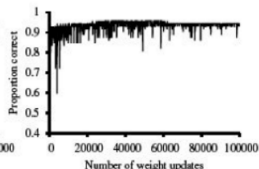
- Pour visualiser la progression de l'apprentissage, on peut regarder la courbe d'apprentissage, c'est-à-dire la courbe du taux d'erreur (ou de succès) en fonction du nombre de mises à jour des paramètres



linéairement
séparable



pas linéairement
séparable



pas linéairement
séparable, avec taux
d'app. décroissant

References I

- ▶ **Hugo Larochelle**, Professeur associé, Université de Montréal, Google
- ▶ **Pierre-Marc Jodoin**, Professeur titulaire Université Sherbrooke
- ▶ Bayesian Reasoning and Machine Learning de David Barber
- ▶ The Elements of Statistical Learning de Trevor Hastie,
- ▶ Robert Tibshirani et Jerome Friedman
- ▶ Information Theory, Inference, and Learning Algorithms de David J.C. MacKay
- ▶ Convex Optimization de Stephen Boyd et Lieven Vandenberghe
- ▶ Natural Image Statistics de Aapo Hyvärinen, Jarmo Hurri et Patrik O. Hoyer
- ▶ The Quest for Artificial Intelligence - A History of Ideas and Achievements de Nils J. Nilsson
- ▶ Gaussian Processes for Machine Learning de Carl Edward Rasmussen et Christopher K. I. Williams
- ▶ Introduction to Information Retrieval de Christopher D. Manning, Prabhakar Raghavan et Hinrich Schütze