



Introduction au Machine Learning

Massih-Reza Amini, Alexandre Audibert, Aurélien Gauffre

Université Grenoble Alpes
Laboratoire d'Informatique de Grenoble
`Prenom.Nom@univ-grenoble-alpes.fr`



Programme

- Perceptron,
- Adaline, Regression Logistique,
- Séparateurs à Vaste Marge (SVM),
- Perceptron Multi-Couches (PMC)

ADaptive LInear NEuron [Widrow & Hoff, 1960]



- Trouver les paramètres du neurone formel en minimisant l'erreur quadratique entre les sorties réelles et les sorties prédites des exemples:

$$\hat{\mathcal{L}}(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^m (y_i - h_{\mathbf{w}}(\mathbf{x}_i))^2$$

- La règle de mise à jour est aussi basée sur l'algorithme de la descente du gradient avec un pas d'apprentissage $\eta > 0$

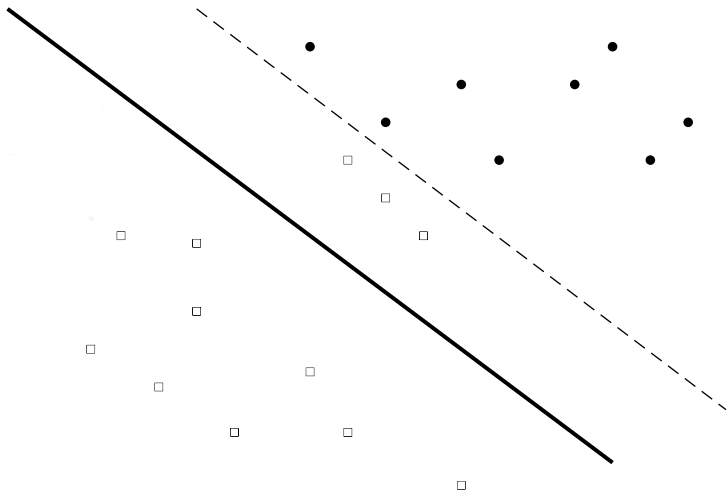
$$\forall (\mathbf{x}, y), \begin{pmatrix} w_0 \\ \bar{\mathbf{w}} \end{pmatrix} \leftarrow \begin{pmatrix} w_0 \\ \bar{\mathbf{w}} \end{pmatrix} + \eta (y - h_{\mathbf{w}}(\mathbf{x})) \begin{pmatrix} 1 \\ \mathbf{x} \end{pmatrix} \quad (1)$$

Pseudo-code d'Adaline

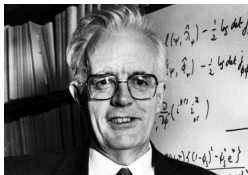
Algorithm 1 Adaline

- 1: Base d'apprentissage $S = \{(\mathbf{x}_i, y_i) \mid i \in \{1, \dots, m\}\}$
 - 2: Initialisation des poids $w^{(0)} \leftarrow 0$
 - 3: $t \leftarrow 0$
 - 4: Pas d'apprentissage $\eta > 0$
 - 5: **repeat**
 - 6: Choisir aléatoirement un exemple $(\mathbf{x}^{(t)}, y^{(t)}) \in S$
 - 7: $w_0^{(t+1)} \leftarrow w_0^{(t)} + \eta \times (y^{(t)} - h_w(\mathbf{x}^{(t)}))$
 - 8: $\bar{w}^{(t+1)} \leftarrow \bar{w}^{(t)} + \eta \times (y^{(t)} - h_w(\mathbf{x}^{(t)})) \times \mathbf{x}^{(t)}$
 - 9: $t \leftarrow t + 1$
 - 10: **until** $t > T$
-

Perceptron vs Adaline



Regression Logistique[Cox, 1969]



- Trouver les paramètres du modèle w en minimisant l'erreur logistique

$$\hat{\mathcal{L}}(w) = \frac{1}{m} \sum_{i=1}^m \ln(1 + e^{-y_i h_w(\mathbf{x}_i)})$$

- La règle de mise à jour avec l'algorithme de la descente de gradient avec un pas d'apprentissage $\eta > 0$

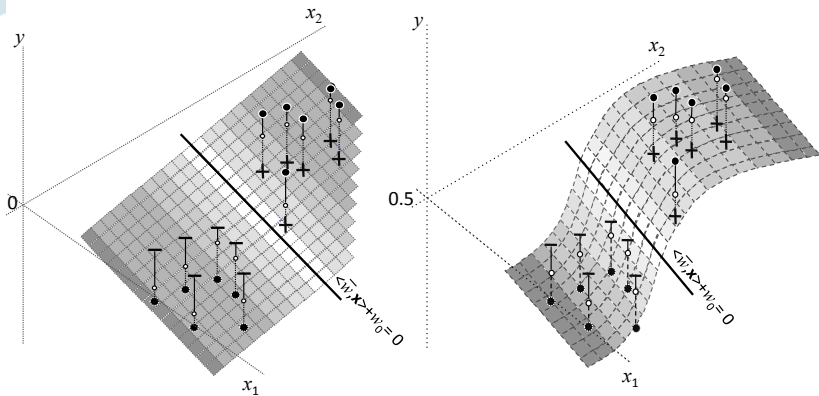
$$\forall(\mathbf{x}, y), \begin{pmatrix} w_0 \\ \bar{w} \end{pmatrix} \leftarrow \begin{pmatrix} w_0 \\ \bar{w} \end{pmatrix} + \eta y (1 - \sigma(h_w(\mathbf{x}))) \begin{pmatrix} 1 \\ \mathbf{x} \end{pmatrix} \quad (2)$$

Pseudo-code Regression Logistique

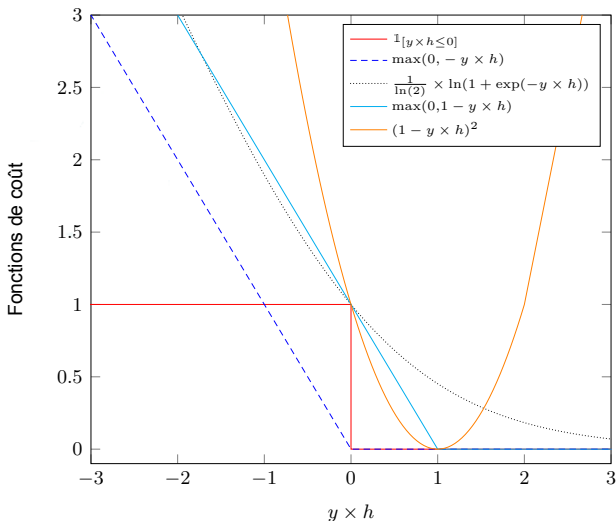
Algorithm 2 Régression Logistique

- 1: Training set $S = \{(\mathbf{x}_i, y_i) \mid i \in \{1, \dots, m\}\}$
 - 2: Initialize the weights $w^{(0)} \leftarrow 0$
 - 3: $t \leftarrow 0$
 - 4: Learning rate $\eta > 0$
 - 5: **repeat**
 - 6: Choose randomly an example $(\mathbf{x}^{(t)}, y^{(t)}) \in S$
 - 7: $w_0^{(t+1)} \leftarrow w_0^{(t)} + \eta \times y^{(t)} (1 - \sigma(h_{w^{(t)}}(\mathbf{x}^{(t)})))$
 - 8: $\bar{w}^{(t+1)} \leftarrow \bar{w}^{(t)} + \eta \times y^{(t)} (1 - \sigma(h_{w^{(t)}}(\mathbf{x}^{(t)}))) \times \mathbf{x}^{(t)}$
 - 9: $t \leftarrow t + 1$
 - 10: **until** $t > T$
-

Adaline vs Régression Logistique

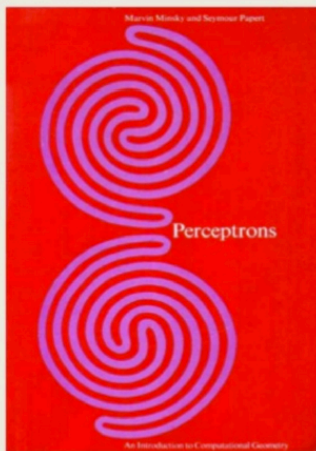


Fonctions de coût vs l'erreur de classification

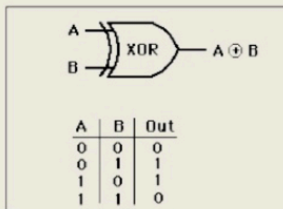


1^{er} hiver des réseaux de neurones

1969: Perceptrons can't do XOR!



<http://www.i-programmer.info/images/stories/BabBag/AI/book.jpg>



<http://hyperphysics.phy-astr.gsu.edu/hbase/electronic/ietron/xor.gif>

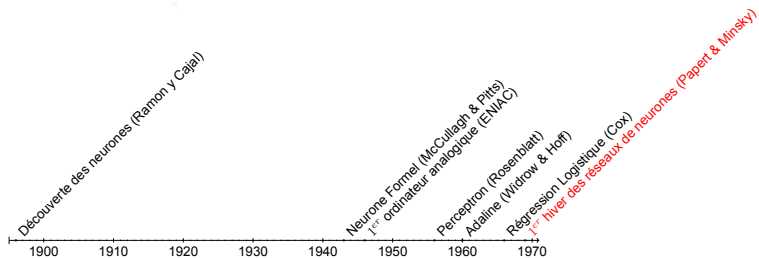


Minsky & Papert

<https://constructingkids.files.wordpress.com/2013/05/minsky-papert-71-csolomon-x640.jpg>

Evolution chronologique

- ❑ Le livre de Minsky et Papert a marqué le 1^{er} hiver des réseaux de neurones;
 - ⇒ **Abandon des algorithmes à base du neurone formel**,
 - ❑ Développement de l'IA symbolique,
 - ❑ À la recherche des modèles non-linéaires



References



G. Widrow and M. Hoff

Adaptive switching circuits.

Institute of Radio Engineers, Western Electronic Show and Convention, Convention Record, 4: 96–104, 1960.



D. Cox

Some procedures connected with the logistic qualitative response curve.

Research Papers in Probability and Statistics (Festschrift for J. Neyman).
London: Wiley. pp. 55–71, 1969.



M. Minsky and S. Papert

Perceptrons: an introduction to computational geometry.

MIT Press, 1969