Faculté des Mathématiques et de l'Informatique

Master : GL 1 Fondements de l'I. A.

Examen Final Corrigé

Questions de cours. (8 pts)

Q1. **Donnez un organigramme expliquant la phase apprentissage et la phase test de l'apprentissage automatique.**

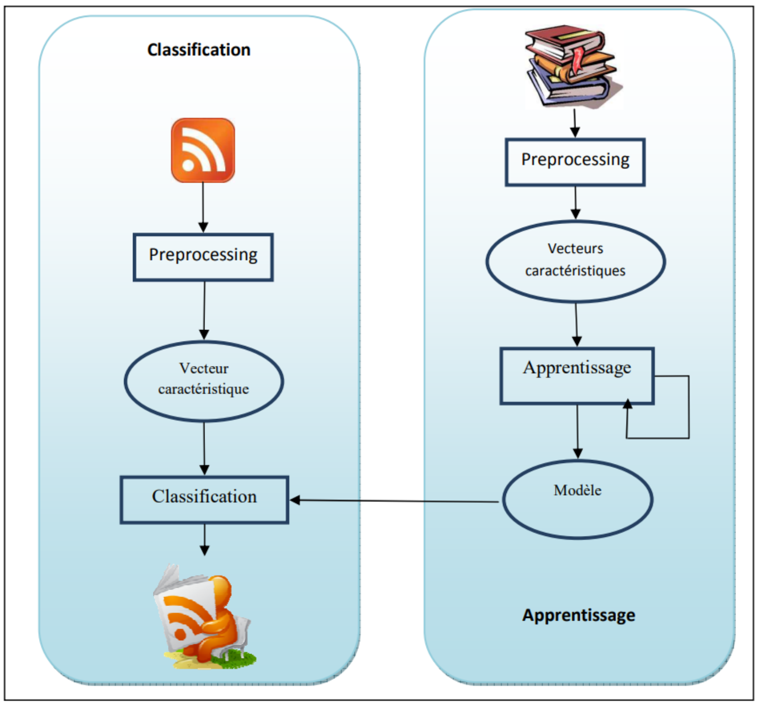


Figure 1: A droite la phase apprentissage et à gauche la phase test de l'apprentissage automatique.

Q2. **Quelle est la forme de la frontière de décision (type de courbe) provenant d'un perceptron ?**

Le perceptron est une structure simple qui prend des entrées, les multiplie par des poids, additionne celles-ci et passe le résultat dans une fonction d'activation (ex : seuil ou sigmoïde).

Selon ce modèle, un perceptron classifiera comme positif un exemple x pour lequel : $w_0+\sum w_i x_i$. Il s'agît donc de l'équation d'un hyperplan dont l'orientation (vecteur normal) est donnée par w et la distance à l'origine par $w_0/\|w\|$.

Un exemple sera classifié selon le côté de l'hyperplan duquel il se situe.

Q3. **Expliquez de manière sommaire l'algorithme d'optimisation basé sur la descente du gradient.**

Le terme de gradient désigne un vecteur représentant la variation d'une fonction par rapport à la variation de ses différents paramètres. Ainsi le gradient d'une fonction convexe f en un point M est le vecteur dont les composantes sont les dérivées partielles de f calculées au point M.

L'algorithme de descente de gradient : partant d'un point sur une surface, on cherche la pente la plus grande en calculant le gradient et on descend d'un petit pas, on recommence à partir du nouveau point jusqu'à atteindre un minimum local.

Q4. **. Expliquez de manière sommaire l'algorithme de rétropropagation pour l'entraînement des réseaux de neurone.**

L'algorithme de retro-propagation sert à trouver les paramètres du réseau de neurones (les poids w_i) qui minimisent l'erreur entre la sortie réelle y_d des exemples d'entraînement x^t , et la sortie y_t produite par le réseau de neurones.

Pour optimiser ces poids, on utilise l'algorithme de la descente du gradient. Dans le cas de réseaux ayant une ou plusieurs couches cachées, la mise-à-jour varie selon la couche du réseau. Ainsi, la modification à apporter à un poids w_h de la couche de sortie correspond à :

$$\triangle w_h = \alpha \sum (y_d - y_t) z_{ht}$$

, où z_{ht} est la sortie du neurone h de la couche cachée précédente.

En revanche, la modification à un poids w_{hj}, connectant l'entrée j au neurone h de la couche cachée, vaut :

$$\triangle w_{hj} = \alpha \sum (y_d - y_t) w_h z_{ht} (1 - z_{ht}) x_{jt}.$$

Comme l'indique son nom, l'algorithme fonctionne en deux phases. Dans un premier temps, les entrées sont propagées vers l'avant pour générer la sortie et son erreur. Ensuite, l'information provenant de l'erreur est propagée vers l'arrière pour mettre à jour les poids.

Ces deux phases sont répétées jusqu'à convergence.

Exercice. (12 pts)

L'apprentissage du perceptron simple a pour objectif la détermination des valeurs des poids synaptiques qui permettent de fournir la sortie (classe) y_i à partir des données d'apprentissage x_i. Autrement dit, il s'agit de déterminer les valeurs des poids synaptiques à partir des paires $x_i, y_i \in D$.

L'algorithme opère de façon itérative pendant t_{max} itérations. Il permet de calculer la prédiction $\sigma(x_i)$ de l'entrée x_i et de la comparer à la classe réelle y_i.

La formule de mise à jour des poids est appelée règle d'apprentissage du perceptron. Si $y_i \neq \sigma(x_i)$, les éléments du vecteur poids w sont mis à jour en fonction de la différence entre les valeurs de y_i et de $\sigma(x_i)$. Cette règle de mise à jour est appelée règle Delta (Δ).

La constante α est appelée pas d'apprentissage ou taux d'apprentissage; elle est souvent fixée à la valeur 0.01.

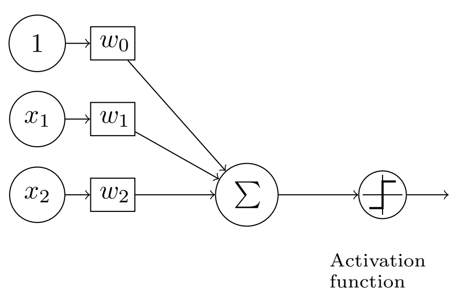
Le critère d'arrêt peut être le nombre maximal d'itérations ou bien une valeur d'erreur fixée au préalable.

Dans l'algorithme d'apprentissage du perceptron suivant, nous avons considéré le nombre maximal d'itérations t_{max} comme critère d'arrêt.

Soit l'ensemble d'apprentissage suivant :

x_0	1	1	1	1	1	1
x_1	0	1	1	1	2	2
x_2	3	1.5	4	1	0.5	2
y	0	0	0	1	1	1

Entrainer le réseau de neurone suivant afin qu'il puisse séparer linéairement les éléments de l'ensemble d'apprentissage : $F(x) = \sigma(w_1x_1 + w_2x_2 + w_0x_0)$



On prendra comme :

1. fonction d'activation sigmoïde $\sigma(t) = \frac{1}{1+\exp^{-t}}$,
2. taux d'apprentissage $\alpha = 1$,

3. initialisation des poids $w = (w_0, w_1, w_2) = (-2, 0, 1)$,
4. L'erreur quadratique moyenne commise par la fonction F associée aux poids : (w_0, w_1, w_2) est : $E(w_0, w_1, w_2) = \frac{1}{N}\sum E_i(w_0, w_1, w_2)$, où N est le nombre total des exemples et $E_i(w_0, w_1, w_2) = \frac{1}{N}(F((x_1, x_2) - z_i)^2$.

Déroulez l'algorithme du perceptron pour deux époques.
Représentez les résultats dans un tableau.

Corrigé :

Algorithm 1 d'apprentissage du perceptron

Require: D, W_0, η
  return W
  $t \leftarrow 1$
  while $t \leq t_{max}$ **do**
    for i:=1 to m step 1 do
    { Tirez un exemple $(x^i, y_i) \in D$
      for j:=1 to d step 1 do
      {
        $\Delta_j \leftarrow \eta x_j(y_t - \sigma(x^i))$
        $w_j \leftarrow w_j + \Delta_j$
      }
    }
    $t \leftarrow t + 1$
  end while

Commençons par traiter le premier exemple $x^1 = (x_0, x_1, x_2) = (1, 0, 3)$:

Les poids initiaux sont $(w_0, w_1, w_2) = (-2, 0, 1)$. On commence par calculer le produit scalaire $\langle w, x\rangle = \langle(-2, 0, 1), (1, 0, 3)\rangle = -2 \times 1 + 0 \times 0 + 1 \times 3 = 1$.

Ensuite,on applique la fonction d'activation $\sigma(1) = 1/(1 + \exp^{-1}) = 0.73$

Appliquons la règle Delta de mise à jour des poids:

$\Delta_0 \leftarrow \eta x_0(y_1 - \sigma(x^1)) = 1 \times 1(0 - 0.73) = -0.73$

$w_0 \leftarrow w_0 + \Delta_0 = -2 - 0.73 = -2.73$

$\Delta_1 \leftarrow \eta x_1(y_1 - \sigma(x^1)) = 1 \times 0(0 - 0.73) = 0$

$w_1 \leftarrow w_1 + \Delta_1 = 0.$

$\Delta_2 \leftarrow \eta x_2(y_1 - \sigma(x^1)) = 1 \times 3(0 - 0.73) = -2.19$

$w_2 \leftarrow w_2 + \Delta_2 = 1 + (-2.19) = -1.19$

Donc les valeurs des poids après traitement du premier exemple sont : $W = (w_0, w_1, w_2) = (-2.73, 0, -1.19)$.
Nous allons utiliser ces nouvelles valeurs des poids W pour traiter le deuxième exemple.
Traitons le deuxième exemple $x^2 = (x_0, x_1, x_2) = (1, 1, 1.5)$:

Les poids courants sont $(w_0, w_1, w_2) = (-2.73, 0, -1.19)$.
La valeur du produit scalaire $\langle w, x\rangle = \langle(-2.73, 0, -1.19), (1, 1, 1.5)\rangle = -4.52$.

Ensuite,on applique la fonction d'activation $\sigma(-4.52.) = 1/(1+\exp^{-4.52.}) = 0.01$

Appliquons la règle Delta de mise à jour des poids:

$\Delta_0 \leftarrow \eta x_0(y_2 - \sigma(x^2)) = 1 \times 1(0 - 0.01) = -0.0107$

$w_0 \leftarrow w_0 + \Delta_0 = -2.73 - 0.010 = -2.741$

$\Delta_1 \leftarrow \eta x_1(y_2 - \sigma(x^2)) = 1 \times 1(0 - 0.01) = -0.01$

$w_1 \leftarrow w_1 + \Delta_1 = -0.0107$.

$\Delta_2 \leftarrow \eta x_2(y_1 - \sigma(x^2)) = 1.5 \times 1(0 - 0.01) = -0.015$

$w_2 \leftarrow w_2 + \Delta_2 = -1.19 - 0.015 = -1.20932$

Donc les valeurs des poids après traitement du deuxième exemple sont : $W = (w_0, w_1, w_2) = (-2.741, -0.0107, -1.20932)$.

Les résultats des deux époques sont reportés dans le tableau suivant :

x^i	y_i	$w_i = (w_0, w_1, w_2)$	$< w_i, x^i >$	$\sigma(< w_i, x^i >)$	Δ_i
(1 0 3)	0	(-2 0 1)	1	0.73	(-0.73105 0 -2.193175)
(1 1 1.5)	0	(-2.731059 0. -1.193176)	-4.52	0.01	(-0.01077 -0.010777 -0.01615)
(1 1 4)	0	(-2.742 -0.0107 -1.209)	-7.59	0.00050	(-0.0005 -0.0005 -0.0020)
(1 1 1)	1	(-2.74 -0.0112 -1.2113)	-3.96	0.02	(0.9813 0.9813 0.98138)
(1 2 0.5)	1	(-1.7609 0.97011 -0.22996)	0.06	0.52	(0.4839 0.9679 0.24199)
(1 2 2)	1	(-1.277 1.9379 0.0120)	2.62	0.93	(0.06767 0.1353 0.1353)
(1 0 3)	0	(-1.2093 2.0733 0.1473)	-0.77	0.32	(-0.31707 -0 -0.9512)
(1 1 1.5)	0	(-1.5264 2.0733 -0.80385)	-0.66	0.34	(-0.34099 -0.34099 -0.5115)
(1 1 4)	0	(-1.8674 1.7323 -1.3153)	-5.4	0.00	(-0.0045 -0.0045 -0.0180)
(1 1 1)	1	(-1.8719 1.7278 -1.3334)	-1.48	0.19	(0.8142 0.8142 0.8142)
(1 2 0.5)	1	(-1.0577 2.5420 -0.5192)	3.77	0.98	(0.0226 0.045 0.0113)
(1 2 2)	1	(-1.0351 2.5872 -0.5079)	3.12	0.96	(0.04215 0.0843 0.08429)

Les valeurs des poids au bout de deux époques sont :

w=(-1.0351 2.5872 -0.5079)+ (0.04215 0.0843 0.08429)

w= (-0.993 2.6715 -0.4236)