

Université d'Avignon

M1 Intelligence Artificielle – Approches Neuronales – TP 1

Juan-Manuel Torres - 2024 - PERCEPTRONS

Travaillez individuellement ou en binôme. Langages acceptés : C/C++ ; python ;
awk, rust, perl ; ruby – Exclus : R, java, javascript
Interdit d'utiliser des fonctions/librairies des RN natives (R, python, matlab, etc)

Théorie

Un perceptron est capable de trouver un hyperplan qui sépare 2 classes d'exemples si l'ensemble est linéairement séparable (LS). Or, créer un ensemble de P points de dimension N (tiré au hasard, dont la classe est tirée au hasard aussi) et garantir qu'il est LS peut s'avérer difficile, car on ne saurait pas dire s'il le sera ou pas avant de le classer par un perceptron... Ceci revient au problème de l'œuf et de la poule.

Une façon de garantir la séparabilité linéaire d'un tel ensemble de données artificielles, consiste à tirer au hasard les poids $\mathbf{W}^*(i)$; $i=0,1,\dots,N$ d'un perceptron appelé **perceptron professeur**, puis d'affecter la classe $\mathbf{TAU}(\mu)$; $\mu=1,\dots,P$; des P points de \mathbf{X} par :

$$\mathbf{TAU}(\mu) = \text{signe} [\mathbf{W}^*(i) \cdot \mathbf{X}(\mu, i)] ; \text{ où } \text{signe}(a) = -1 \text{ si } a < 0, +1 \text{ autrement}$$

Travail pratique

0/ Perceptron. Programmer l'algorithme du perceptron a) version batch et b) version online. Pour chaque version faire apprendre un perceptron \mathbf{W} sur les ensembles d'apprentissage suivants :

- i/ fonction ET entrées binaires ($N=2$) ;
- ii/ fonction OU entrées binaires ($N=2$) ;
- iii/ Un des ensemble au choix des exemples vus au cours.

Fixer le taux d'apprentissage **eta** à une valeur adéquate $0 < \text{eta} < 1$
Vérifier que l'algorithme marche bien.

1/ Données LS aléatoires. Construire un ensemble LS de P exemples en $N+1$ dimensions avec un **perceptron professeur** \mathbf{W}^* . Attention : le poids $\mathbf{W}^*(0)$ étant le biais, donc $\mathbf{X}(\mu, 0)=1$

2/ Apprentissage. Apprendre avec a) la **version batch** de l'algorithme du perceptron et b) la **version online**. Pour chaque version faire apprendre un **perceptron élève** \mathbf{W} sur l'ensemble LS obtenu lors de la génération de données. Calculer :

- a) Les $N+1$ poids \mathbf{W} du perceptron élève ;
 - b) Le nombre d'itérations IT nécessaires pour converger ;
 - c) Le recouvrement R entre le perceptron professeur \mathbf{W}^* et l'élève \mathbf{W} :
- $$R = \cos [(\mathbf{W}^* \cdot \mathbf{W}) / (\| \mathbf{W}^* \| \cdot \| \mathbf{W} \|)] ; \quad \| \mathbf{a} \| = \text{norme de } \mathbf{a} = \sqrt{\sum a(i)^2} ; i=0,1,\dots,N$$

3/ Tests. Lancer le programme avec les valeurs: N=2,10,100,500,1000,5000; P=10,100,500,1000. Donner vos résultats sous la forme de 2 tableaux (batch et online) où chaque case affiche la moyenne du nombre d'itérations **<IT>** et la moyenne du rapport **<R>**, sur 50 tirages aléatoires.

Construisez trois tableaux en fonction de eta : d'abord l'eta0 de l'exercice 0, puis avec eta=eta0/2 et finalement eta=eta0/10

eta

	P=10	P=100	P=500	P=1000
N = 2	<IT>;<R>			
N = 10				
N = 100				
N = 5000		

eta/2

	P=10	P=100	P=500	P=1000
N = 2	<IT>;<R>			
N = 10				
N = 100				
N = 5000		

eta/10

	P=10	P=100	P=500	P=1000
N = 2	<IT>;<R>			
N = 10				
N = 100				
N = 5000		

4/ Questions. Que pouvez vous dire des moyennes de **<IT>** et de **<R>** par rapport à N et P ? Que pouvez dire du temps d'exécution en fonction d'eta pour la version batch et celle online ?

5/ Rapport. Tableaux de moyennes + réponse aux questions. Rendu PDF et codes sources compressés ZIP/GZIP. dépôt sur ENT.