Université d'Avignon
M1 Intelligence Artificielle – Apprentissage supervisé – TP 1
Juan-Manuel Torres - 2025 - PERCEPTRONS

Travaillez individuellement ou en binôme. Langages acceptés : C/C++ ; python ; awk, rust, perl ; ruby – Exclus : R, java, javascript
<u>Interdit d'utiliser des fonctions/librairies des RN natives (R, python, matlab, etc).</u>
<u>Interdit d'utiliser IA pour générer vos codes, par contre IN à volonté</u>

<u>Théorie</u>

L'algorithme du perceptron peut trouver un hyperplan qui sépare 2 classes d'exemples si l'ensemble d'apprentissage 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 s'avére difficile, car on ne saurait pas dire s'il est ou pas LS 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 **W***(i); i=0,1,...,N d'un perceptron appelé **<u>perceptron professeur</u>**, puis d'affecter la classe **TAU**(mu); mu=1,...,P ; des P points de **X** par :

TAU(mu) = signe [**W***(i) · **X**(mu, i)] ; où **signe(a)** = -1 si a < 0, +1 autrement

<u>Nous allons étudier certaines propriétés de ces ensembles aléatoires</u>

<u>Travail pratique</u>

0/ Perceptron. Programmer l'algorithme du perceptron a) version batch et b) version online. Pour chaque version faire apprendre un perceptron 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 < **eta** < 1. Vérifiez que l'algorithme marche bien.

1/ Données LS aléatoires. Construire un ensemble LS de P exemples en N+1 dimensions avec un **<u>perceptron professeur</u>** **W*.** *Attention : le poids* **W***(0) *étant le biais, donc* **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 ensemble de 20 **<u>perceptrons élèves</u>** **W** sur l'ensemble LS obtenu lors de la génération de données. Calculez :

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

Utilisez **threads** pour l'apprentissage des elèves.

3/ Tests. Lancer le programme avec les valeurs: N=2,10,100,500,1000,5000; P=10,100,200,500,1000. Donner vos résultats sous la forme de 2 tableaux (ou graphiques en couleurs = bonus) (batch et online) où chaque case affiche la <u>moyenne</u> du nombre d'itérations **<IT>** et la <u>moyenne</u> du rapport **<R>**, sur 100 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
Bonus : utilisez threads pour vos tests

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. <u>Rendu PDF</u> et codes sources compressés ZIP/GZIP. dépôt sur ENT.