

Projet I.A. 2^{ème} partie

Question 1 (1 point)

Qu'est-ce que le Q-learning ? Présentez un exemple d'application (1 page maximum).

Question 2, arbre de décision (1 point)

On considère les données suivantes :

Variables :

Type (type de mission) : courte / longue

Prop (propulsion interplanétaire) : chimique / solaire électrique / nucléaire électrique

Astro (nombre d'astronautes) : entre 3 et 6

MOI (Mars orbit insertion) : aérocapture / freinage propulsif

ERV (Earth return vehicle) : assemblage du véhicule du retour en orbite terrestre / martienne

EDL (technique d'entrée, descente et atterrissage) : bouclier rigide / bouclier gonflable

10 missions ont été proposées et évaluées TC « trop complexe » ou A « acceptable ».

Propositions de mission :

	Type	Prop	Astro	MOI	ERV	EDL	Eval.
1	courte	chimique	6	freinage	terrestre	gonflable	TC
2	courte	solaire	6	freinage	terrestre	gonflable	TC
3	longue	chimique	3	aérocapt.	martienne	rigide	A
4	longue	solaire	4	freinage	terrestre	gonflable	A
5	courte	nucléaire	6	freinage	terrestre	gonflable	TC
6	longue	chimique	6	freinage	terrestre	gonflable	TC
7	longue	chimique	4	aérocapt.	terrestre	gonflable	A
8	longue	solaire	6	freinage	martienne	gonflable	TC
9	courte	nucléaire	3	aérocapt.	terrestre	rigide	TC
10.	courte	chimique	3	aérocapt.	martienne	gonflable	TC

Construire un arbre de décision permettant de déterminer si la mission est trop complexe ou acceptable en fonction des 6 paramètres de la mission. L'arbre doit être aussi petit que possible. Présentez les calculs et le raisonnement qui ont conduit au résultat.

Question 3 : classification avec Perceptron 1 couche (1 point)

On considère les données fournies dans le fichier texte « A et B.txt ». 30 mesures ont été faites pour la population A et 30 mesures ont été faites pour la population B.

Soit un perceptron 1 couche comportant 3 entrées : (x,y) parmi les données et une constante égale à 1) et une sortie (voir cours pour le calcul de la sortie du neurone). Programmez l'apprentissage du perceptron avec les données fournies afin que la sortie du perceptron soit 1 si et seulement si le couple de données en entrée provient de la population A et 0 sinon.

Rappel de l'algorithme :

Initialisation des poids $W1$, $W2$ et $W3$ de manière aléatoire.

Tant qu'il existe une erreur de classification et qu'on n'a pas effectué 1000000 itérations faire

Initialisation à 0 du nombre d'erreurs de classification

Pour chacun des couples (E1,E2) de la base de données

Initialiser les entrées

Calculer la sortie

Si la sortie vaut 0 alors que 1 était attendu, $W_i \leftarrow W_i + E_i$ pour chaque W_i

Si la sortie vaut 1 alors que 0 était attendu $W_i \leftarrow W_i - E_i$ pour chaque W_i

Si la sortie n'est pas la bonne, augmenter de 1 le nombre d'erreurs

Augmenter de 1 le nombre d'itérations

Afficher le nombre d'erreurs et les valeurs finales des poids

NB : alternez les exemples, prenez-en 1 de A puis 1 de B et ainsi de suite, sinon il risque d'y avoir un biais.

A rendre :

- Votre programme
- Une copie d'écran de votre programme permettant de visualiser notamment les valeurs initiales et finales des poids, le nombre d'itérations et le nombre d'erreurs.
- Un graphe montrant la position des points et la droite séparatrice définie par les poids.

Question 4 : régression avec un MLP (2 points)

On considère le fichier de données « donneespb3.txt », qui résulte d'un test effectué en interférométrie optique.

Dans ce fichier, à chaque ligne est indiqué un couple de réels (x,y) correspondant à une position sur un écran et une valeur de z indiquant une mesure d'intensité lumineuse entre 0 et 1. Il y a en tout 3000 mesures. On souhaite utiliser un perceptron multi-couches (Multi-Layer Perceptron) pour apprendre automatiquement, grâce à la méthode de rétropropagation du gradient de l'erreur, l'intensité en fonction de (x,y).

Utilisez les fichiers Csharp fournis pour écrire votre propre programme réalisant cet apprentissage.

Présentez votre résultat sous la forme de 2 fichiers image de taille 500x500 pixels, le premier correspondant approximativement aux données fournies et le 2^{ème} à la sortie du perceptron pour tous les couples d'entiers (x,y) de l'image. NB : ré-étalonnez les valeurs d'intensité entre 0 et 255, puis affectez la valeur (arrondie) aux composantes RVB du pixel pour obtenir un dégradé de gris.

Discutez ensuite de la qualité du résultat et de la vitesse de convergence de l'apprentissage en fonction des paramètres suivants :

- Nombre de neurones sur la couche cachée.
- Coefficient d'apprentissage.
- Taux d'erreur résiduel (par exemple la moyenne des erreurs entre chaque sortie désirée et chaque sortie obtenue).

Question subsidiaire : quelle est la fonction mathématique approximativement encodée par le perceptron ?

Question 5 : classification (5 points)

On considère les données du fichier donneespb4.txt. Les 2 premières valeurs réelles (x,y) correspondant à un point dans le plan et la 3^{ème} indique la classe d'appartenance : si c'est 0.2 c'est la première classe et si c'est 0.8 c'est la 2^{ème} classe.

L'objectif est de tester l'apprentissage supervisé et l'apprentissage non supervisé sur ces données.

1. Apprentissage supervisé

Nous allons utiliser un perceptron multi-couches (voir fichiers Csharp fournis). Réalisez un programme qui permet d'effectuer un apprentissage supervisé et totalement automatisé de la classification de ces points. Le résultat doit être obtenu en 1 seul clic. Pour visualiser le résultat de la classification après apprentissage, adoptez la méthode suivante : faites passer tous les couples (x,y) correspondant aux pixels d'une image de taille 500x500 pixels en entrée du perceptron, puis lancez le calcul de la sortie et coloriez les pixels en bleu ou en rouge selon la sortie/classe obtenue. Ensuite, par-dessus, affichez les points correspondant aux données du fichier (valeurs arrondies), ceux de la classe A en blanc et ceux de la classe B en noir.

L'objectif est d'obtenir le meilleur résultat possible en un minimum de temps. Vous chercherez donc à optimiser votre technique d'apprentissage par toute adaptation que vous jugerez utile (boosting...).

Discutez ensuite de la qualité des résultats et de la vitesse de convergence en fonction des paramètres suivants :

- Nombre de neurones sur la couche cachée.
- Coefficient d'apprentissage.
- Taux d'erreur résiduel (par exemple la moyenne des erreurs entre chaque sortie désirée et chaque sortie obtenue).
- Traitement complémentaire ajouté ou pas, par exemple pour le boosting.

2. Apprentissage non supervisé

Pour l'apprentissage non supervisé, on utilisera une carte autoorganisatrice, l'algorithme de Kohonen et une technique de regroupement permettant d'aboutir, in fine, à un nombre de classes fixé à l'avance (voir programme fourni). Pour réaliser un apprentissage, il faut :

- a) Recueillir l'ensemble E des points de la base de données.
- b) Créer une carte auto-organisatrice, procéder à un apprentissage non supervisé selon la technique de Kohonen, puis effectuer un regroupement pour obtenir plusieurs classes. Notez que, puisqu'il s'agit d'un apprentissage non supervisé, l'algorithme ne peut pas exploiter l'information d'appartenance de chaque point. Néanmoins, on peut combiner cette approche avec de l'apprentissage supervisé en exploitant cette information pour une ultime étape de classification, notamment lorsque des sous-ensembles de points de la même classe sont totalement disjoints.
- c) On teste ensuite la classification en colorant les points selon la classification obtenue et on affiche également le pourcentage de bien classés et de mal classés.

Comme pour l'apprentissage supervisé, l'objectif est d'obtenir le meilleur résultat possible. Vous chercherez donc à optimiser votre technique d'apprentissage et à procéder à des tests pertinents.

Discutez ensuite des résultats obtenus en fonction, notamment :

- Du nombre de neurones de la carte.
 - De la vitesse de convergence.
 - Du nombre de classes demandées pour le regroupement.
 - De l'ajout d'une ultime étape de regroupement supervisé.
-

A rendre lors du dernier TP d'avril :

- Les programmes réalisés : perceptron 1 couche ; perceptron multi-couches pour question 4 ; perceptron multi-couches et carte SOM pour question 5.
- Dans le rapport, une présentation de vos éventuels ajouts, les résultats obtenus et les réponses aux questions.

Nota Bene : Indiquez de manière détaillée la répartition des tâches au sein de votre groupe. En cas de trop grande disparité, une note différente pourra être attribuée à chaque élève.