

# TD Intelligence Artificielle

## Exercice 1 :

Supposons que l'on a un problème de classification qui consiste à déterminer la classe d'appartenance de chacune des instances. Le domaine de valeurs des classes possibles est : {1, 2, 3}.

Instance	A1	A2	A3	A4	A5	Classe
X1	3	5	4	6	1	1
X2	4	6	10	3	2	2
X3	8	3	4	2	6	3
X4	2	1	4	3	6	3
X5	2	5	1	4	8	2

Selon la base de connaissance précédente, déterminez la classe de l'instance X6, dont les valeurs pour les attributs A1 à A5 sont < 3, 12, 4, 7, 8 >, à l'aide de l'algorithme des k-voisins les plus proches. Montrez tous les calculs et précisez la distance utilisée.

## Exercice 2 :

Pour faire des diagnostics A, B ou C, des médecins se basent sur une analyse morphologique des cellules bactériennes prélevées dans le sang de leurs patients. Après avoir observé les douze échantillons suivants :

Numéro	Maladie	Nbr noyaux	Nbr flagelles	Coloration	Paroi
1	A	1	1	Pâle	Fine
2	A	2	1	Pâle	Fine
3	A	1	1	Pâle	Épaisse
4	A	1	1	Foncée	Fine
5	A	1	1	Foncée	Épaisse
6	B	2	2	Pâle	Fine
7	B	2	2	Foncée	Fine
8	B	2	2	Foncée	Épaisse
9	C	2	1	Foncée	Fine
10	C	2	1	Foncée	Épaisse

11	C	1	2	Pâle	Fine
12	C	1	2	Pâle	Épaisse

Ils aimeraient construire l'arbre de décision le plus simple possible pour classer leurs instances.

Montrez comment cet arbre peut être construit en appliquant l'algorithme CART. Donnez uniquement les 2 premiers nœuds de cet arbre en indiquant tous vos calculs.

On rappelle que la fonction de Gini est donnée par :

$$Gini(p) = 1 - \sum_{k=1}^c P^2(k|p) = 2 \sum_{k < k'} P(k|p)P(k'|p)$$

### Exercice 3 :

On veut classifier en 3 groupes les 8 données qui figurent dans la matrice des données ci-dessous à gauche. Une partie de la matrice de distances est reproduite ci-dessous à droite.

id	$(x, y)$		A	C	D
A	(-3, -2)	A	0.00	6.08	5.83
B	(-2, 0)	B	2.24	5.10	3.61
C	( 3, -1)	C	6.08	0.00	5.00
D	( 0, 3)	D	5.83	5.00	0.00
E	( 3, 0)	E	6.32	1.00	4.24
F	( 2, -1)	F	5.10	1.00	4.47
G	(-3, -3)	G	1.00	6.32	6.71
H	( 0, 4)	H	6.71	5.83	1.00

On utilise l'algorithme k-means initialisé sur les points A, C et D.

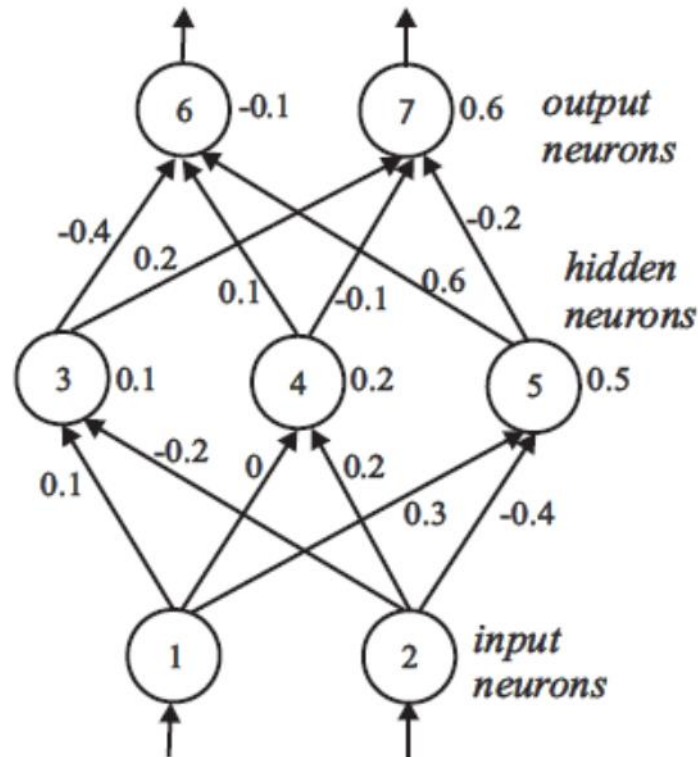
1. Donner les trois centres après la première itération.
2. Donner les trois groupes résultant de la classification (Représenter graphiquement les données afin d'éviter les calculs).

### Exercice 4 :

Étant donné le réseau de neurones suivant avec les poids initialisés comme dans l'image, expliquez l'architecture de réseau en sachant que nous essayons de distinguer les clous et les vis et un exemple d'entraînement de tuples est le suivant : T1 {0.6, 0.1, clou}, T2 {0.2, 0.3, vis}

Avec le taux d'apprentissage égal à 0,1 et les poids comme indiqué dans la figure. Effectuez la propagation en avant des signaux dans le réseau en

utilisant T1 comme entrée, puis effectuez la propagation en arrière de l'erreur. Montrer les changements des poids.



### Exercice 5 :

Considérons un réseau de neurones qui reçoit en entrée des images en teintes de gris représentées par des matrices de  $20 \times 20$  pixels.

Chacune de ces images illustre un des fruits suivants : banane, pomme, poire ou ananas.

Il s'agit donc d'un problème de classification multi-classes, le réseau possède quatre neurones de sorties.

1. Si le réseau possède une seule couche cachée pleinement connectée de  $n$  neurones, combien de paramètres devons-nous optimiser par descente de gradient ?
2. Si le réseau possède une première couche de  $m$  filtres de convolutions de taille  $5 \times 5$ , suivie d'une couche cachée pleinement connectée de  $n$  neurones, et d'aucun paramètre de biais, combien de paramètres devons-nous optimiser par descente de gradient ?
3. Si le réseau possède une première couche de  $m$  filtres de convolutions de taille  $5 \times 5$ , suivit d'une couche de *MaxPooling* avec une taille de pas

de 4, suivit d'une couche cachée pleinement connectée de  $n$  neurones, et d'aucun paramètre de biais, combien de paramètres devrons-nous optimiser par descente de gradient ?

4. Enfin, considérons l'architecture suivante :

- Entrée du réseau : images de tailles  $20 \times 20$  ;
- 1<sup>ère</sup> couche :  $m1$  filtres de convolutions de taille  $5 \times 5$ , sans biais.
- *MaxPooling* avec taille de pas 2
- 2<sup>e</sup> couche :  $m2$  filtres de convolutions de taille  $3 \times 3 \times m1$ , sans biais.
- *MaxPooling* avec taille de pas 2
- 3<sup>e</sup> couche :  $n1$  neurones pleinement connectés, sans biais
- 4<sup>e</sup> couche :  $n2$  neurones pleinement connectés, sans biais
- 5<sup>e</sup> couche : 4 neurones de sortie, avec biais.