**UNIVERSITE CADDI AYYAD**
**FACULTE DES SCIENCES SEMLALIA**
**INFORMATIQUE LPI S5, 17 Janvier 2008**

# Contrôle : Intelligence Artificielle

**Durée 2 h**

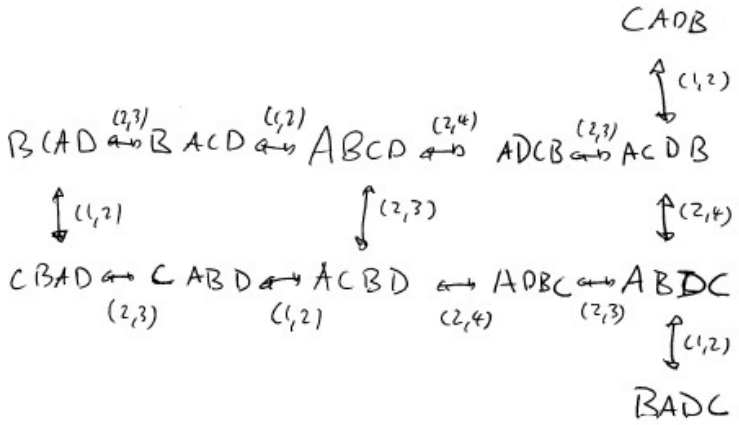
**Cours :**

1. Donner et expliquer l'algorithme Largeur d'abord (complétude, optimalité, complexité en temps et espace).
2. Qu'est ce que l'apprentissage (le Quoi, le pourquoi et le comment) ?.
3. Expliquer l'apprentissage non supervisé.

**Exercice1**

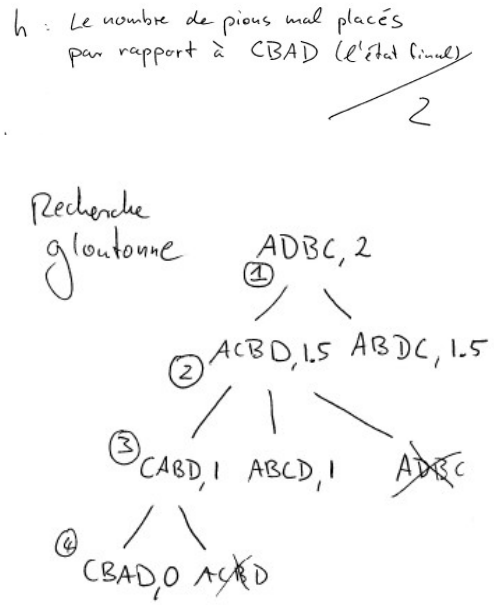
Nous considérons un monde avec 4 pions **(A,B,C,D)** non superposables. Ils peuvent être arrangés dans n'importe quel ordre, sauf **A** qui ne peut pas être plus à droite que **D**. Par exemple, **ABCD** et **CBAD** sont deux états possibles du monde, tandis que **DCBA** et **CDAB** ne sont pas possibles. Le monde peut être manipulé par une action de la forme ***echange(x, y)*** qui échange les pions des positions x et y. Par exemple ***echange(1, 2)*** transforme **BCAD** dans **CBAD**. Seules les actions ***echange(1, 2)***, ***echange(2, 3)*** et ***echange(2, 4)*** sont autorisées. Ils donnent un successeur uniquement si la situation atteinte est possible.

• Dessinez le graphe d'états.

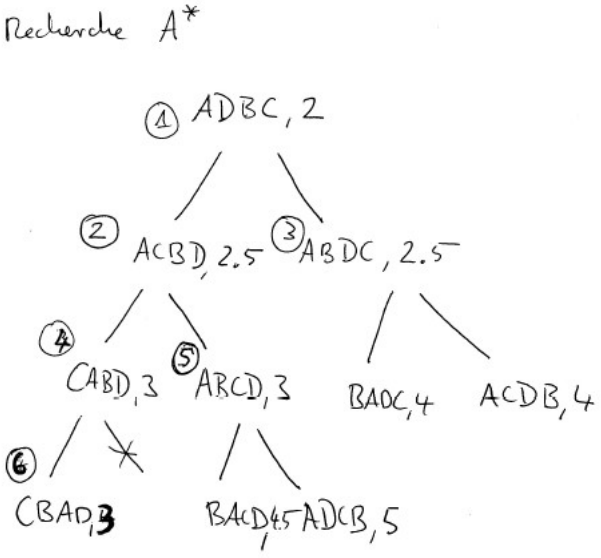


• On suppose que l'état de départ est **ADBC** et l'état que l'on veut atteindre est **CBAD**. On suppose que chaque action coûte **1**. Donnez une "bonne" heuristique h admissible (mais aussi différente de **0** pour les noeuds non-finaux) pour ce problème. Le principe de l'heuristique devrait être suffisamment général pour pouvoir s'appliquer `a des problèmes similaires.

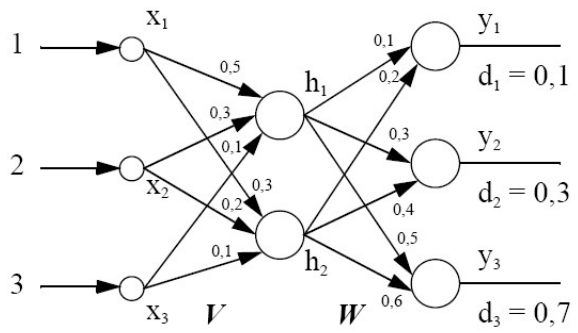
• Appliquez la recherche gloutonne avec votre heuristique. Si vous n'avez pas trouvé d'heuristique, utilisez l'heuristique **h = (nombre de pions mal placés)**. Ne considérez pas les noeuds déjà développés. En cas d'égalité choisissez un nœud à développer au hasard.




• Appliquez la recherche A* avec votre heuristique. Si vous n'avez pas trouvé d'heuristique, utilisez l'heuristique **h = (nombre de pions mal placés)/2**. Ne considérez pas les noeuds déjà développés. En cas d'égalité choisissez un nœud à développer au hasard.




**Exercice2**
Soit le perceptron multicouche suivant :




Dans l'unique but de simplifier les calculs, les neurones ne sont pas munis de l'habituel paramètre de polarisation (seuil). Les poids de connexion affichés directement sur la connexion sont résumés dans les deux matrices de connexion :

$$\boldsymbol{V} = \begin{bmatrix} 0,5 & 0,3 & 0,1 \\ 0,3 & 0,2 & 0,1 \end{bmatrix} \qquad \boldsymbol{W} = \begin{bmatrix} 0,1 & 0,2 \\ 0,3 & 0,4 \\ 0,5 & 0,6 \end{bmatrix}$$

Calculez les nouvelles valeurs de poids des matrices de connexion $\boldsymbol{V}$ et $\boldsymbol{W}$ après une passe complète de propagation directe - rétropropagation du gradient.
Les paramètres du réseau sont :

$$\eta = 1 \qquad f(net) = \frac{1}{1 + e^{-net}}$$

net = somme pondérée au niveau d'une cellule.

$$\text{le stimulus } \mathbf{x} = \begin{bmatrix}1\\2\\3\end{bmatrix} \quad \text{doit donner la réponse} \quad \mathbf{t} = \begin{bmatrix}.1\\.3\\.7\end{bmatrix}$$

Cette activation est ensuite convertie en réponse. En utilisant la fonction logistique, on obtient:

$$\mathbf{h} = f(\mathbf{b}) = \begin{bmatrix}0.8022\\0.7311\end{bmatrix} . \qquad \textbf{(VI.15)}$$

Cette activation est ensuite transmise aux cellules de la couche de sortie. Elles calculent leur activation:

$$\mathbf{a} = \mathbf{Z}\mathbf{h} = \begin{bmatrix}0.2264\\0.5331\\0.8397\end{bmatrix} , \qquad \textbf{(VI.16)}$$

elles la transforment en réponse en utilisant la fonction logistique:

$$\mathbf{o} = f(\mathbf{a}) = \begin{bmatrix}0.5564\\0.6302\\0.6984\end{bmatrix} . \qquad \textbf{(VI.17)}$$

Les cellules de sortie peuvent évaluer leur *signal d'erreur*:

$$\boldsymbol{\delta}_{\text{sortie}} = f'(\mathbf{a}) \circledast \mathbf{e} = \mathbf{o} \circledast (\mathbf{1} - \mathbf{o}) \circledast (\mathbf{t} - \mathbf{o})$$
$$= \begin{bmatrix}-0.1126\\-0.0770\\0.0003\end{bmatrix} .$$

Réponse :

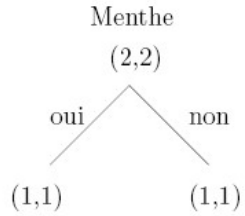
$$V = \begin{bmatrix}0,4946 & 0,2892 & 0,0837\\0,2896 & 0,1791 & 0,0687\end{bmatrix} \qquad W = \begin{bmatrix}0,0096 & 0,1177\\0,2383 & 0,3437\\0,5003 & 0,6002\end{bmatrix}$$

**Exercice 3.**
Les chercheurs de la chaîne de café Columbus ont collecte les informations suivantes concernant le fait si les clients aiment leur café avec différents arômes ajoutes. Les trois attributs sont des attributs binaires qui indiquent si l'arôme a été ajoute ou pas.

| Menthe | Noisette | Vanille | Aimé ? |
|---|---|---|---|
| oui | oui | non | non |
| oui | non | non | oui |
| non | non | non | oui |
| non | oui | non | non |

- Donner l'attribut à la racine de l'arbre de décision avec ID3. Donnez les détails du calcul. Est-ce qu'après avoir choisi la racine on doit choisir un autre noeud? Pourquoi ?

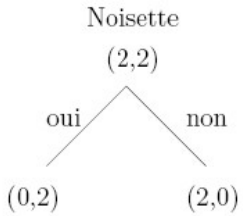
**On considère les trois arbres de décision donnés par les trois tests possibles.**



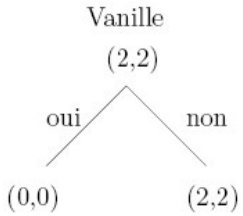

$Entropie(\epsilon) = -((2/4)log_2(2/4) + (2/4)log_2(2/4)) = 1$

$Entropie(1) = -((1/2)log(1/2) + (1/2)log(1/2)) = 1$

$Entropie(2) = -((1/2)log(1/2) + (1/2)log(1/2)) = 1$
$Gain(\epsilon) = Entropie(\epsilon) - ((1/2)Entropie(1) + (1/2)Entropie(2)) = 0$




$Entropie(\epsilon) = 1$ **(c'est la même que pour Menthe)**
$Entropie(1) = -((0/2)log(0/2) + (2/2)log(2/2)) = 0$
$Entropie(2) = -((2/2)log(2/2) + (0/2)log(0/2)) = 0$
$Gain(\epsilon) = Entropie(\epsilon) - ((1/2)Entropie(1) + (1/2)Entropie(2)) = 1$




$Entropie(\epsilon) = 1$ **(c'est la même que pour Menthe)**
$Entropie(1) = 0$ **(il n'y a aucun exemple)**
$Entropie(2) = -((2/4)log(2/4) + (2/4)log(2/4)) = 1$
$Gain(\epsilon) = Entropie(\epsilon) - ((0 * Entropie(1) + 1 * Entropie(2)) = 0$

**On choisit donc le test Noisette. On voit que s'est suffisant parce qu'on donner une classe aux feuilles sans faire une faute.**

- Est-ce que avec un perceptron linéaire à seuil on peut aussi classifier correctement l'échantillon ? Justifiez (Donnez le perceptron ou montrez qu'il n'y en a pas).

**Oui. On prends un perceptron à trois entrées (correspondant aux attributs Menthe, Noisette et Vanille) avec les poids $(w_1, w_2, w_3) = (0, -1, 0)$ et le seuil $-0.5$.**

- De manière général, est-ce que pour un échantillon avec des attributs binaires et deux classes associées on peut toujours trouver un arbre de décision parfait ? Sous quelles conditions ? Est-ce qu'on peut toujours trouver un perceptron qui classifie correctement ? Sous quelles conditions ?

Pour un échantillon avec des attributs binaires et deux classes associées on peut toujours trouver un arbre de décision parfait à condition qu'il n'y ait pas de contradiction (deux exemple avec les mêmes valeurs des attributs mais deux classes différentes). Il y a un perceptron qui classifie correctement si l'échantillon est linéairement séparables.

**log2=1, log1/2=-1**
**(w0,w1,w2,w3) = (-0.5,0,-1,0)**