b. L'algorithme C4.5

- Prise en compte les attributs numérique
- Des attributs dont l'arité est élevée (voire infinie)
- C'est un successeur d'*ID3*
- Dans le cas de *C4.5*, un nœud de l'arbre de décision peut contenir un test du fait que la valeur d'un attribut numérique est inférieure à un certain seuil : cela correspond donc à un nouveau pseudo-attribut binaire. *C4.5* ne dispose pas d'autres possibilités de prendre en compte ce type d'attributs.

Exemple

Nous illustrons le déroulement de l'algorithme *C4.5*sur le jeu de données « jouer au tennis ? » dans lequel les attributs «Température» et «Humidité» prennent des valeurs numérique

Jour	Ciel	Température	Humidité	Vent	Jouer Au Tennis?
1	Ensoleillé	27.5	85	Faible	Non
2	Ensoleillé	25	90	Fort	Non
3	Couvert	26.5	86	Faible	Oui
4	Pluie	20	96	Faible	Oui
5	Pluie	19	80	Faible	Oui
6	Pluie	17.5	70	Fort	Non
7	Couvert	17	65	Fort	Oui
8	Ensoleillé	21	95	Faible	Non
9	Ensoleillé	19.5	70	Faible	Oui
10	Pluie	22.5	80	Faible	Oui
11	Ensoleillé	22.5	70	Fort	Oui
12	Couvert	21	90	Fort	Oui
13	Couvert	25.5	75	Faible	Oui
14	Pluie	20.5	91	Fort	Non

- Test d'un attribut numérique

Considérons les exemples dont l'attribut « *Ciel* » vaut « Ensoleillé », soit l'ensemble X_{Ciel}=ensoleille d'exemples ayant un seul attribut numérique comme suit :

Ministère de l'Enseignement supérieur et de la recherche scientifique
Université Abderrahmane Mira de Bejaïa
Faculté des Sciences Exactes
Département de Mathématiques

Module: Data Mining **Année: 2022-2023**
Nature de document: Cours
Niveau :L3-STID **N.BERMAD**
Chapitre: 2(suite)
Titre: Apprentissage supervisée et Réseaux de neurones

Jour	Température	"Jouer au Tennis"
1	27.5	Non
2	25	Non
8	21	Non
9	19.5	Oui
11	22.5	Oui

On commence par trier les exemples sur la valeur de leur attribut numérique. A chaque attribut, on associe le numéro de son exemple associé ainsi que la valeur de l'attribut cible :

Température	19.5	21	22.5	25	27.5
Jour	9	8	11	2	1
"Jouer au tennis?"	Oui	Non	Oui	Non	Non

On détermine le seuil S pour partitionner cet ensemble d'exemples. $C4.5$ utilise les règles suivantes :

1. Ne pas séparer deux exemples successifs ayant la même classe ; donc, on ne peut couper qu'entre les exemples x_9 et x_8, x_8 et x_{11}, x_{11} et x_2 .

2. Si on coupe entre deux valeurs V et W $(V < W)$ de l'attribut, le seuil S est fixé à V (on aurait pu aussi utiliser $V+W/2$)
3. Choisir S de telle manière que le gain d'information soit maximal.
4. Une fois le seuil S fixé et le nœud créé, chaque sous-arbre pourra à nouveau tester la valeur de cet attribut
5. En effet, contrairement au cas des attributs qualitatifs qui produisent des nœuds ayant autant de branches que l'attribut prend de valeurs différentes, l'ensemble des valeurs prises par un attribut numérique est coupé en deux : chaque partie peut donc encore être raffinée jusqu''a ne contenir que des exemples ayant même valeur cible.

L'entropie de l'ensemble d'exemples est :

$$E(C_{jouer = oui}, C_{jouer = non}) = -(\frac{2}{5} ln2 \frac{2}{5} + \frac{3}{5} ln2 \frac{3}{5}) \cong 0.971$$

Pour $S = 21$, le gain d'information est :

Gain en information de température :

$$Gain(température) = 0.971 - (1/5 * E(C_{température<21}) + 4/5 * E(C_{température>=21}))$$

avec:

$$E(C_{température<21}) = -(1ln_2 1 + 0ln_2 0) = 0$$

et

$$E(C_{température>=21}) = -(1/4ln_2 1/4 + 3/4ln_2 3/4) \cong 0.608$$

soit

$$gain(C,S=21) \cong 0.971 - (\frac{1}{5} * 0 + \frac{4}{5} * 0.608) \cong 0.485$$

de la même manière, en fonction du seuil, le gain d'information est alors:

seuil	Gain(C, température,S)
S=21	0.485
S=22.5	0.02
S=25	0.42

Ministère de l'Enseignement supérieur et de la recherche scientifique
Université Abderrahmane Mira de Bejaïa
Faculté des Sciences Exactes
Département de Mathématiques

Module: Data Mining **Année: 2022-2023**
Nature de document: Cours
Niveau :L3-STID **N.BERMAD**
Chapitre: 2(suite)
Titre: Apprentissage supervisée et Réseaux de neurones

6. *C4.5* effectue ce traitement pour chaque attribut quantitatif et détermine donc pour chacun un seuil produisant un gain d'information maximal.
7. Le gain d'information associé à chacun des attributs quantitatifs est celui pour lequel le seuil entraine un maximum.
8. Finalement, l'attribut choisi (parmi les quantitatifs et les nominaux pour lesquels le principe est identique *ID3*) est celui qui produit un gain d'information maximal
9. En présence d'attribut numérique ou d'attribut d'arité élevée, ceux-ci sont automatiquement favorisés pour être sélectionné comme test dans les nœuds. Pour contrecarrer cet effet, *C4.5* utilise le rapport de gain au lieu du gain d'information pour déterminer l'attribut `a utiliser dans un nœud.

Le rapport de gain est défini par :$Rapportdegain(C,A) = \frac{Gain(C,A)}{SplitInfo(C,A)}$

où

$$SplitInfo(C,A) = \sum_{v \in valeurs(A)} \frac{|CA = v|}{25} ln2 \frac{|CA = v|}{|C|}$$

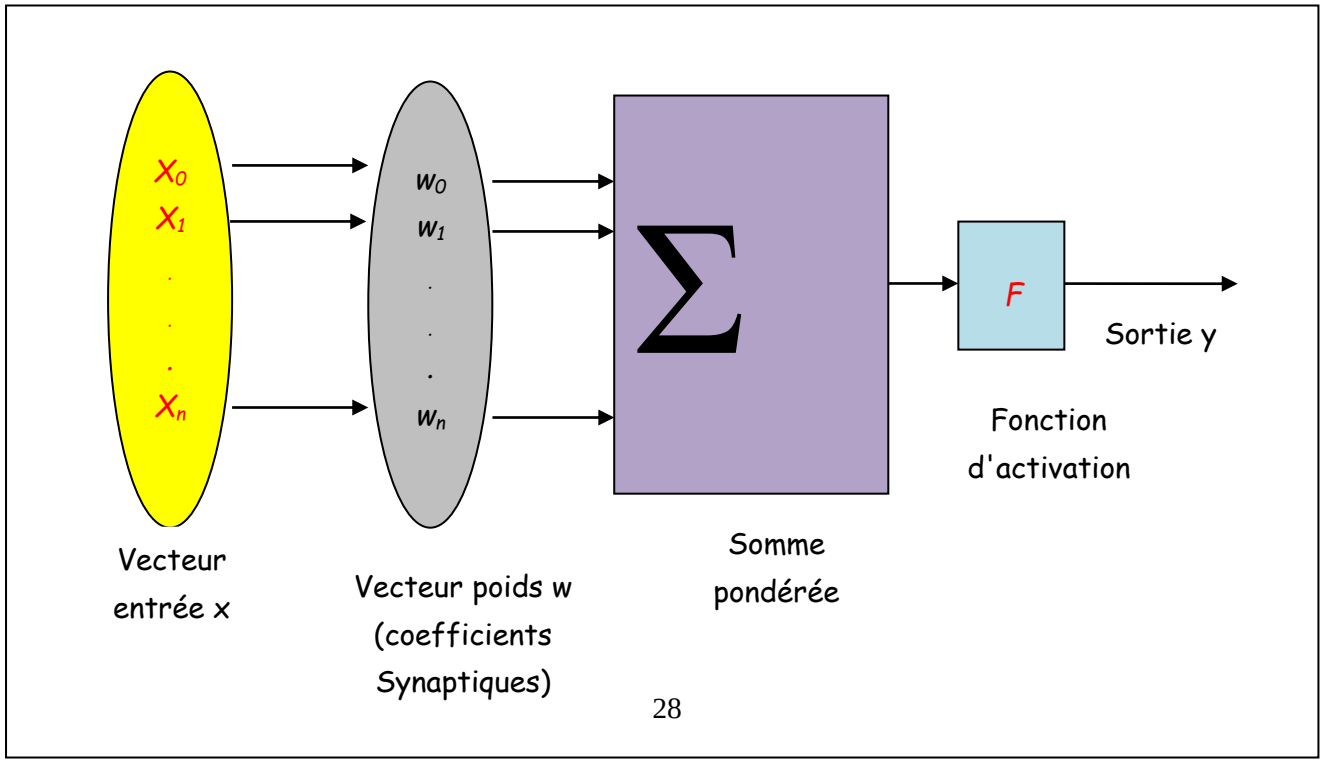
Arbres de décision avantages & inconvénients

+ Compréhensible pour tout utilisateur (lisibilité du résultat-règles-arbre)
+Justification de la classification d'une instance
+Tout type de données
+ Robuste au bruit et aux valeurs manquantes
+Attributs apparaissent dans l'ordre de pertinence
+ Classification rapide (parcours d'un chemin dans un arbre)
+Outils disponible dans la plupart des environnements de data mining.
- Sensible au nombre de classes: performances se dégradent
-Evolution dans le temps: si les données évoluent dans le temps, il est nécessaire de relance la phase d'apprentissage.

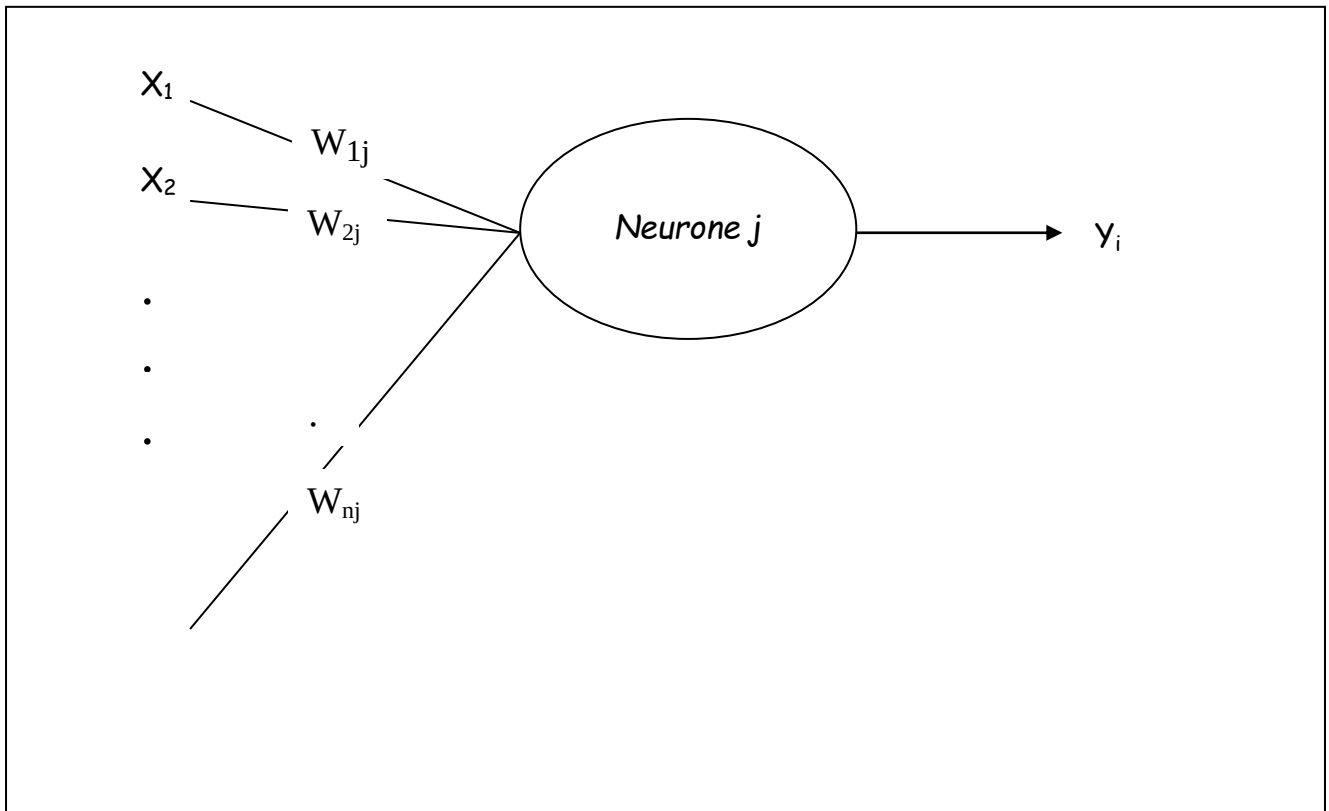
- **2.5.4. Les réseaux de neurones artificiels**
- **Réseau neuronal :**
  - Simule le système nerveux biologique

  - Un réseau de neurones est composé de plusieurs neurones ou d'unité de traitement interconnectés entre elles. Un poids est associé à chaque arc. A chaque neurone on associe une valeur
  - Les réseaux de neurones se distinguent en particulier par la topologie des connexions et le mécanisme d'apprentissage
- **Neurone:**
  - Unité de calcul élémentaire
  - Le vecteur d'entrée X est transformé en une variable de sortie Y, par un produit scalaire et une fonction de transformation non linéaire

27


X_0
X_1
X_n
w_0
w_1
w_n
Σ
F
Sortie y
Fonction d'activation
Vecteur entrée x
Vecteur poids w (coefficients Synaptiques)
Somme pondérée

28


X_1
X_2
W_{1j}
W_{2j}
W_{nj}
Neurone j
y_i

X_n

- Le neurone prend un ensemble de valeur Xi en entrée et produit une sortie Yi calculée de la manière suivante:

$$yi = f(\sum_{i=1}^{n} xiwij + biaisj) \text{ OÙ } f \text{ est une fonction d'activation}$$

le biais peut être assimilé à un poids W_{0j} pour une entrée X_0 qui est toujours activées(X_0=1).Voici quelques exemples de fonctions d'activation populaires:

- Linéaire: $f(x) = ax$
- Seuil: $f(x) = 1 \; si \; x > 0 \; sinon \; 0$
- Sigmoïde (logistique): $f(x) = 1/1 + e^{-x}, donc \; f(x) \in [0..1]$

- Certains mécanismes d'apprentissage exigent que la fonction d'activation soit dérivable, ce qui n'est pas le cas de la fonction seuil. La sortie y peut être passée à un autre neurone qui effectue le même calcul à son tour

2.5.4.1.Mise en œuvre d'un réseau

- Les étapes dans la mise en œuvre d'un réseau de neurones pour la prédiction ou le classement sont :
- Identification des données en entrée et en sortie
- Normalisation de ces données
- Constitution d'un réseau avec une topologie adaptée
- Apprentissage du réseau
- Test du réseau
- Application du modèle généré par l'apprentissage
- Dé normalisation des données en sortie.

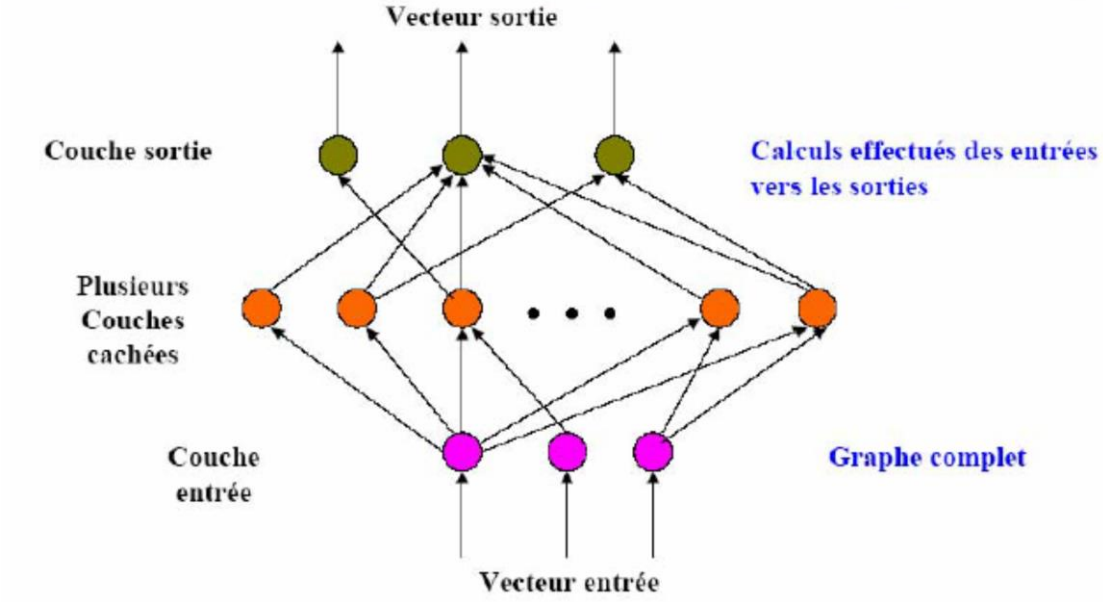
2.5.4.2. Modèles de réseau de neurones:

- Le **réseau à fonction radiale de base** (« radial basis function » *RBF*) est aussi utilisé pour prédire une variable cible continue ou discrète
- Le **réseau de Kohonen** effectue les analyses typologiques (clustering, recherche de segments)
- etc...

Perceptron multicouches:

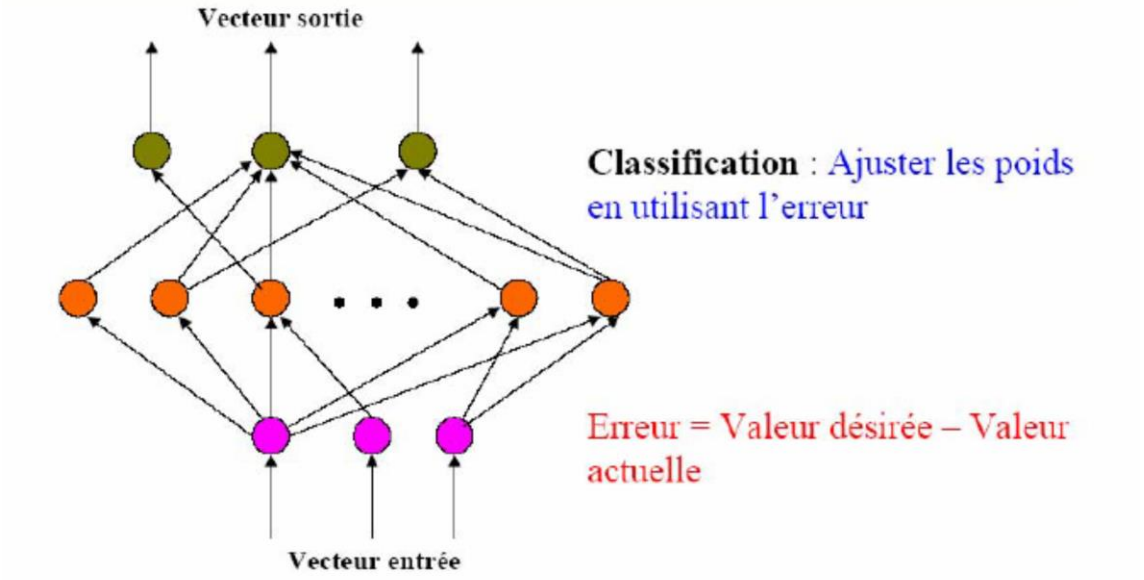
- Est utilisé pour prédire une variable cible continue ou discrète

Module: Data Mining **Année: 2022-2023**
Nature de document: Cours
Niveau :L3-STID **N.BERMAD**
Chapitre: 2(suite)
Titre: Apprentissage supervisée et Réseaux de neurones



32

Paradigme d'apprentissage



33

Ministère de l'Enseignement supérieur et de la recherche scientifique
Université Abderrahmane Mira de Bejaïa
Faculté des Sciences Exactes
Département de Mathématiques

Module: Data Mining **Année: 2022-2023**
Nature de document: Cours
Niveau :L3-STID **N.BERMAD**
Chapitre: 2(suite)
Titre: Apprentissage supervisée et Réseaux de neurones

Algorithme d'apprentissage :Rétro-propagation du gradient

- **Construction de réseau:**
  - Représentation des entrées
  - Nombre de nœuds en entrée: correspond à la dimension des données du problème (attributs ou leurs codages) et normaliser dans l'intervalle [0,1].

Exemple: un attribut A prends ses valeurs {1, 2, 3, 4,5}

- 5 entrée à la valeur binaire; 3=00100
- 3 bit; 3=010
- 1 entrée réelle;0,0.25, 0.5, 0.75, 1

  - Nombre de couches cachées: ajuster pendant l'apprentissage.
  - Nombre de nœuds par couche: le nombre de nœuds par couche est aux moins égale à deux et au plus égal au nombre de nœuds en entrée
  - Nombre de nœuds en sortie: fonction du nombre de classes associées à l'application.

- **Apprentissage de réseau utilisant les données disponibles**
  - Pour effectuer la classification, le réseau doit être entrainé en ajustant les poids (incluant les biais) à partir de l'échantillon d'entrainement
  - Au départ les poids peuvent être initialisés de manière quelconque
  - Chacun des éléments de l'échantillon est donné en entrée et la sortie est comparée avec la classe d'appartenance de l'élément
  - L'apprentissage est effectué en ajustant les poids en fonction de la différence entre la réponse produite et la bonne réponse
  - Un mécanisme populaire d'ajustement des poids est la rétro-propagation de l'erreur, le principe est d'ajuster les poids de manière à rapprocher la réponse du système de celle attendue

- En appliquant le principe d'optimisation bien connu de descente du gradient pour minimiser la moyenne des carrés d'erreurs, on obtient les formules suivantes d'ajustement des poids

$$\boldsymbol{wij = wij + \Delta wij}$$

$$\boldsymbol{\Delta wij = vitesseApprentissage * Erreurj * yi}$$

$$\boldsymbol{Erreurj = yi(1 - yi)(valeur\ attendu - yi)} \quad \boldsymbol{pour\ neurone\ j\ en\ sortie}$$

$$\boldsymbol{= yi(1 - yi)\sum Erreurkwjk} \quad \boldsymbol{pour\ neurone\ j\ de\ la\ couche\ cachée}$$

- La vitesse d'apprentissage est un paramètre ajustable dont la valeur typique entre 0 et 1.On peut la faire décroitre graduellement afin de stabiliser l'évolution des poids. Par exemple, en prenant une valeur 1/n où n croît avec le nombre d'élément traités
- pour le biais, les formules d'apprentissage sont:
- $biaisij = biaisij + \Delta biaisij$
- $\Delta biaisij = vitesseApprentissage * Erreurj$
- L'apprentissage peut être effectué en ligne en ajustant les paramètres à chacun des patrons d'entrée ou en lot en accumulant les ajustements pour un lot de patrons
- d'entrée. Dans l'apprentissage en lot, chacun des cycles d'apprentissage appelé une époque
- le critère d'arrêt: la tolérance définit l'erreur cible ou/et le nombre d'instances bien classés (seuil)

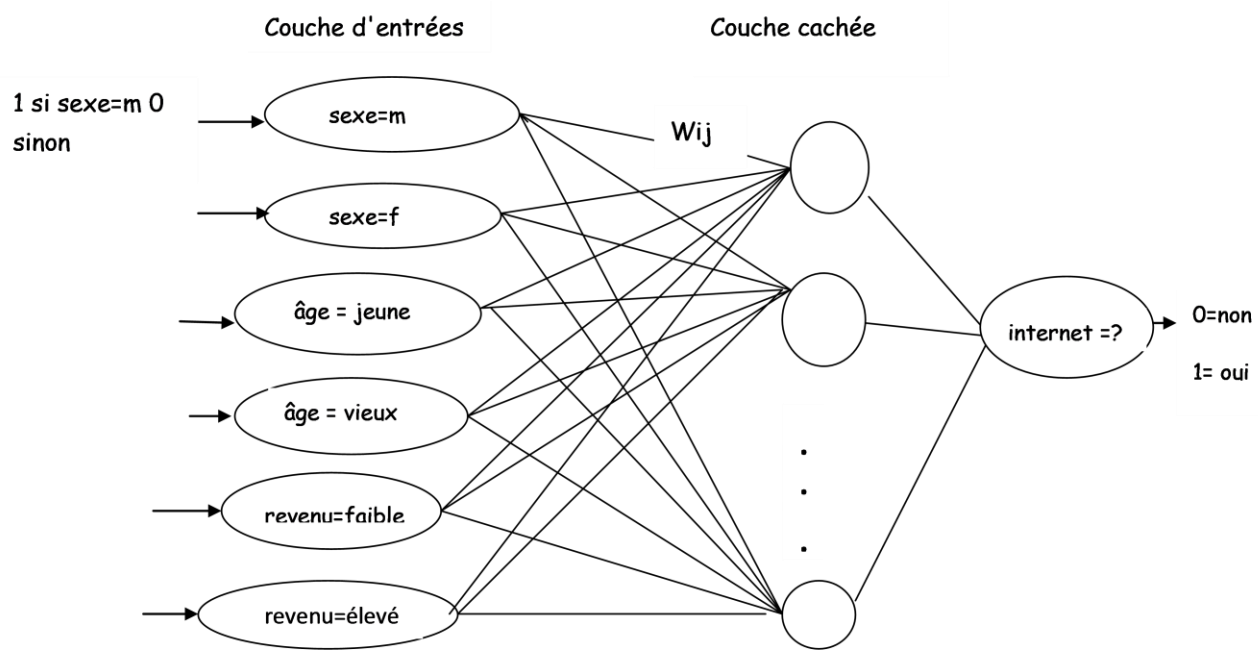
- **Elagage de réseau**

- Réseau fortement connexe est difficile à articuler: N nœuds en entrée, h couches cachées, et m nœuds en sortie → $h(m+n)\ arcs(poids)$
- Elagage: supprimer les arcs et les nœuds qui n'affectent pas le taux d'erreur du réseau

- **Interprétation des résultats**

Exemple

La figure ci dessus montre un exemple de RNA pour notre problème de classification des données de profils Internet. Le réseau prend en entrée les données qui représentent un élément à classifier et produit en sortie une réponse qui identifie la classe d'appartenance de l'élément. Chaque ovale représente un neurone artificiel. Un arc représente une connexion entre deux neurones et W_{ij} est le poids de la connexion du neurone i au neurone j .Les arcs sont implicitement orientés de gauche à droite dans notre diagramme. La sortie d'un neurone est passée comme entrée à un autre neurone ou encore à la sortie. Ce réseau est constitué de trois couches, une couche d'entrée, une couche cachée et une couche de sortie. La couche d'entrée représente les données utilisées pour la classification. Il ya un neurone pour chacun des valeurs d'un attribut. Pour représenter le fait qu'un élément possède une valeur particulière d'attribut, une valeur 1 sera transmise en entrée au neurone et la valeur 0 sera transmise pour les autres neurones du même attribut. Par convention, la valeur 1 en sortie du seul neurone de la couche de sortie représente la classe internet=oui et un 0, la valeur internet=non.



Réseaux de neurones - Avantages

- Taux d'erreur généralement bon
- Outil disponible dans les environnements de data mining
- Robustesse (bruit)-reconnaissance de formes (son, images sur une rétine,...)
- Classification rapide (réseau étant construit)
- Combinaison avec d'autres méthodes (ex:arbre de décision pour sélection d'attributs)

Réseaux de neurones - Inconvénients

- Apprentissage très long
- Plusieurs paramètres (architecture, coefficients synaptique, ...)
- Pouvoir explicatif faible (boite noire)
- Pas facile d'incorporer les connaissances du domaine.
- Traitent facilement les attributs numériques et binaires
- Evolutivité dans le temps (phase d'apprentissage)

Ministère de l'Enseignement supérieur et de la recherche scientifique
Université Abderrahmane Mira de Bejaïa
Faculté des Sciences Exactes
Département de Mathématiques

Module: Data Mining **Année: 2022-2023**
Nature de document: Cours
Niveau :L3-STID **N.BERMAD**
Chapitre: 2(suite)
Titre: Apprentissage supervisée et Réseaux de neurones