

## Les réseaux de neurones

Antoine Cornuéjols  
(antoine@lri.fr)

I.I.E.  
&  
L.R.I., Université d'Orsay

### 2. Historique (très rapide)

- ❑ Prémisses
  - McCulloch & Pitts (1943) : 1er modèle de neurone formel.  
Rapport neurone et calcul logique : base de l'intelligence artificielle.
  - Règle de Hebb (1949) : apprentissage par renforcement du couplage synaptique
- ❑ Premières réalisations
  - ADALINE (Widrow-Hoff, 1960)
  - PERCEPTRON (Rosenblatt, 1958-1962)
  - Analyse de Minsky & Papert (1969)
- ❑ Nouveaux modèles
  - Kohonen (apprentissage compétitif, ...)
  - Hopfield (1982) (réseau bouclé)
  - Perceptron Multi-Couches (1985)
- ❑ Analyse et développements
  - Théorie du contrôle, de la généralisation (Vapnik, ...)

### Le Perceptron Multi-Couches : propagation

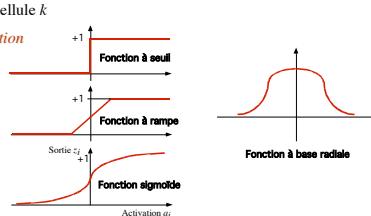
- Pour chaque neurone :

$$y_l = g\left(\sum_{j=0,d} w_{jk} \phi_j\right) = g(a_k)$$

- ❑  $w_{jk}$  : **poids** de la connexion de la cellule  $j$  à la cellule  $k$
- ❑  $a_k$  : **activation** de la cellule  $k$
- ❑  $g$  : **fonction d'activation**

$$g(a) = \frac{1}{1 + e^{-a}}$$

$$g'(a) = g(a)(1-g(a))$$



### Introduction : Pourquoi les réseaux de neurones ?

#### • Inspiration biologique

- ❑ Le cerveau naturel : un modèle très séduisant
  - Robuste et tolérant aux fautes
  - Flexible. Facilement adaptable
  - S'accommode d'informations incomplètes, incertaines, vagues, bruitées ...
  - Massivement parallèle
  - Capable d'apprentissage

#### ❑ Neurones

- $\approx 10^{11}$  neurones dans le cerveau humain
- $\approx 10^4$  connexions (synapses + axones) / neurone
- Potentiel d'action / période réfractaire / neuro-transmetteurs
- Signaux excitateurs / inhibiteurs

### Introduction : Pourquoi les réseaux de neurones ?

#### • Les attraits pratiques

- ❑ Calculs parallélisables
- ❑ Implantables directement sur circuits dédiés
- ❑ Robustes et tolérants aux fautes (calculs et représentations distribués)
- ❑ Algorithmes simples
- ❑ D'emploi très général

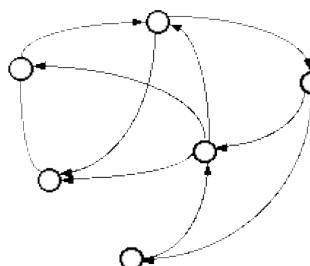
#### • Les défauts

- ❑ Opacité des "raisonnements"
- ❑ Opacité des résultats

### Les réseaux de neurones : Types de réseaux

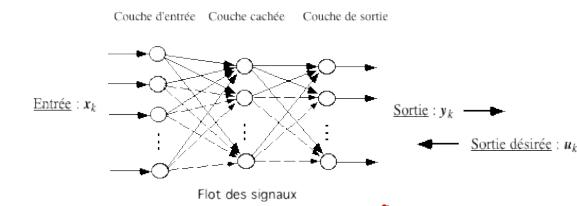
#### • Interconnecté à boucles (e.g. réseau de Hopfield)

- ❑ Fonctionnement en reconnaissance
- ❑ Apprentissage ?

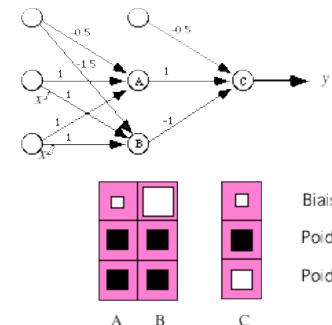


### Modèles de base : le Perceptron Multi-Couches

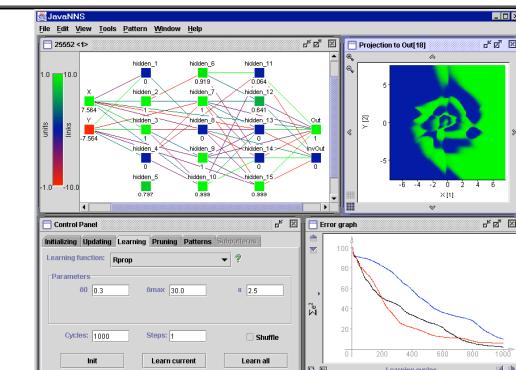
#### • Topologie typique



### Le Perceptron Multi-Couches : exemple du XOR



### Exemple de réseau (simulateur JavaNNS)



## Le PMC : l'apprentissage

- Trouver des poids permettant au réseau de réaliser une relation entrée-sortie spécifiée par des exemples de cette relation

(Toujours le problème de la généralisation)

### Apprentissage :

- Minimiser la fonction de coût  $E(w, \{x^l, u^l\})$  en fonction du paramètre  $w$

- Utiliser pour ceci une **méthode de descente de gradient**

$$\Delta w_{ij} \propto -\partial E / \partial w_{ij}$$

(algorithme de **rétro-propagation de gradient**)

- Principe inductif :** On fait alors l'hypothèse que ce qui marche sur les exemples (minimisation du risque empirique), marche sur des données non vues (minimisation du risque réel)

## Le Perceptron Multi-Couches : apprentissage

Objectif :  $w^* = \arg \min_w \sum_{l=1}^m [y(x_l; w) - u(x_l)]^2$

- Algorithme (rétro-propagation de gradient) : descente de gradient

$$w^{(t)} = w^{(t-1)} - \eta \nabla_E w^{(t)}$$

Cas hors-ligne (gradient total) :  $w_{ij}(t) = w_{ij}(t-1) - \eta(t) \frac{1}{m} \sum_{k=1}^m \frac{\partial R_E(x_k, w)}{\partial w_{ij}}$

où :  $R_E(x_k, w) = [t_k - f(x_k, w)]^2$

Cas en-ligne (gradient stochastique) :  $w_{ij}(t) = w_{ij}(t-1) - \eta(t) \frac{\partial R_E(x_k, w)}{\partial w_{ij}}$

## PMC : La rétro-propagation de gradient

1. Evaluation de l'erreur  $E^l$  (ou  $E$ ) due à chaque connexion :  $\frac{\partial E^l}{\partial w_{ij}}$

Idée : calculer l'erreur sur la connexion  $w_{ij}$  en fonction de l'erreur **après** la cellule  $j$

$$\frac{\partial E^l}{\partial w_{ij}} = \frac{\partial E^l}{\partial a_j} \frac{\partial a_j}{\partial w_{ij}} = \delta_j z_i$$

- Pour les cellules de la **couche de sortie** :

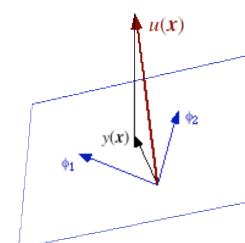
$$\delta_k = \frac{\partial E^l}{\partial a_k} = g'(a_k) \frac{\partial E^l}{\partial y_k} = g'(a_k) \cdot (u_k(x_l) - y_k)$$

- Pour les cellules d'une **couche cachée** :

$$\delta_j = \frac{\partial E^l}{\partial a_j} = \sum_k \frac{\partial E^l}{\partial a_k} \frac{\partial a_k}{\partial a_j} = \sum_k \delta_k \frac{\partial a_k}{\partial z_j} \frac{\partial z_j}{\partial a_j} = g'(a_j) \cdot \sum_k w_{jk} \delta_k$$

## L'apprentissage : Erreur quadratique

### Interprétation géométrique



## Le Perceptron Multi-Couches : apprentissage

### 1. Présentation d'un exemple parmi l'ensemble d'apprentissage

Séquentielle, aléatoire, en fonction d'un critère donné

### 2. Calcul de l'état du réseau

### 3. Calcul de l'erreur = $fct(\text{sortie} - \text{sortie désirée})$ (e.g. $= (y^l - u^l)^2$ )

### 4. Calcul des gradients

Par l'algorithme de rétro-propagation de gradient

### 5. Modification des poids synaptiques

### 6. Critère d'arrêt

Sur l'erreur. Nombre de présentation d'exemples, ...

### 7. Retour en 1

## L'apprentissage : descente de gradient

- Apprentissage = recherche dans l'espace multidimensionnel des paramètres (poids synaptiques) en vue de minimiser la fonction de coût

- Quasitotalité des règles d'apprentissage pour les RNs

### = méthode de descente de gradient

□ Solution optimale  $w^*$  tq. :  $\nabla E(w^*) = \mathbf{0}$

$$\nabla = \left[ \frac{\partial}{\partial w_1}, \frac{\partial}{\partial w_2}, \dots, \frac{\partial}{\partial w_N} \right]^T$$

$$E^{(t+1)} = E^{(t)} - \nabla_w E$$

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} - \eta \frac{\partial E}{\partial w_{ij}} \Big|_{w^{(t)}}$$

## PMC : La rétro-propagation de gradient

## PMC : La rétro-propagation de gradient

### 2. Modification des poids

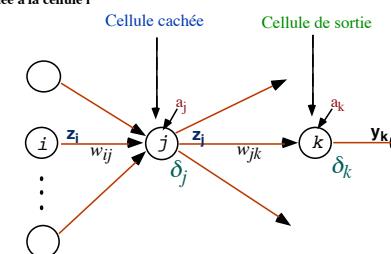
- On suppose gradient à pas (constant ou non) :  $\eta(t)$

- Si **apprentissage stochastique** (après présentation de chaque exemple)

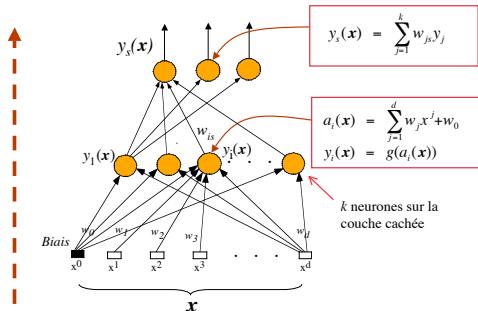
$$\Delta w_{ji} = \eta(t) \delta_j a_i$$

- Si **apprentissage total** (après présentation de l'ensemble des exemples)

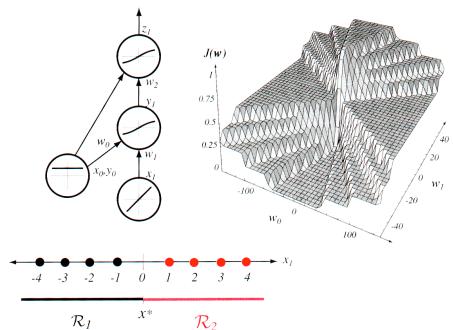
$$\Delta w_{ji} = \eta(t) \sum_n \delta_j^n a_i^n$$



## Le PMC : passes avant et arrière (résumé)



## Analyse de la surface d'erreur



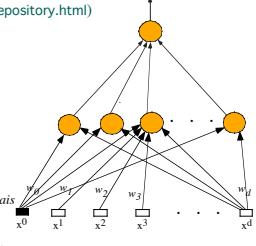
## Applications : la discrimination

- Exemple :

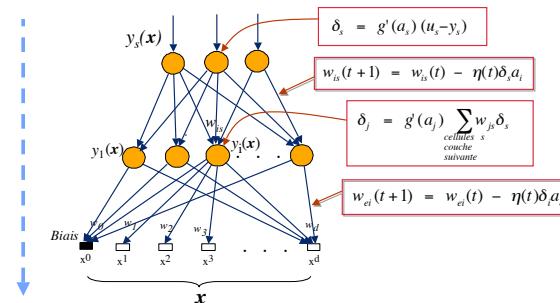
- Mines cylindriques / roches  
(<http://www.ics.uci.edu/mlrepository.html>)

- 1 neurone de sortie

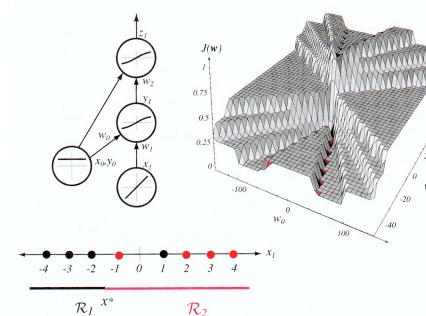
- {0,1}
- [0,1]
  - Erreur quadratique
- Probabilité [0,1]
  - Critère entropique



## Le PMC : passes avant et arrière (résumé)



## Analyse de la surface d'erreur



## Applications : la discrimination multiclasse

- Exemple :

- Reconnaissance de caractères manuscrits
- Reconnaissance de locuteurs

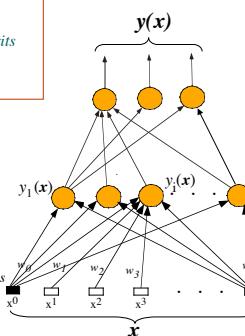
- c-1 problèmes de discrimination

- 1 neurone de sortie

- {0,1, ..., c}
- [0,1]

- n ( $\leq$  c) neurones de sortie

- 1 neurone / classe
- Code correcteur d'erreur



## PMC : La rétro-propagation de gradient

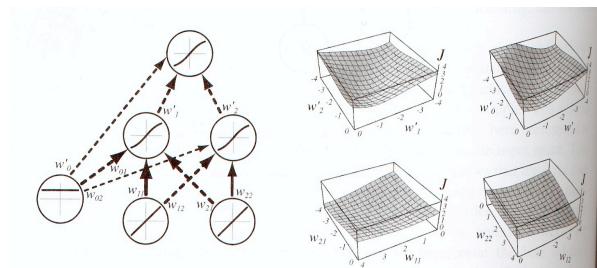
- Efficacité en apprentissage

- En O(w) pour chaque passe d'apprentissage,  $w$  = nb de poids
- Il faut typiquement plusieurs centaines de passes (voir plus loin)
- Il faut typiquement recommencer plusieurs dizaines de fois un apprentissage en partant avec différentes initialisations des poids

- Efficacité en reconnaissance

- Possibilité de temps réel

## Analyse de la surface d'erreur



## Applications : optimisation multi-objectif

- cf [Tom Mitchell]

**PMC : Les applications**

- Automatique** : identification et contrôle de processus (e.g. Commande de robot)
- Traitement du signal** (filtrage, compression de données, traitement de la parole (Identification du locuteur, ...))
- Traitement d'images, reconnaissance des formes** (reconnaissance de l'écriture manuscrite, [Lecture automatique des codes postaux](#) (Zip codes, USA, ...))
- Prédiction** (consommations d'eau, d'électricité, météorologie, bourse, ...)
- Diagnostic** (industrie, médecine, science, ...)

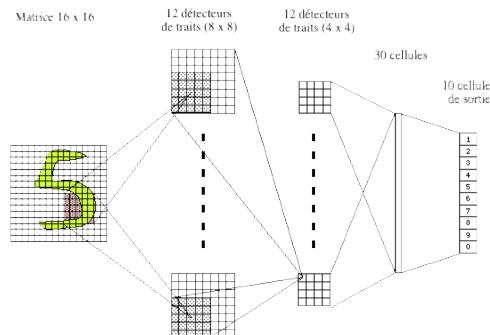
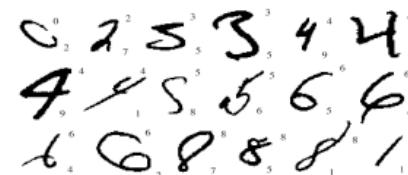
**Application aux codes postaux (Zip codes)****Les erreurs commises**

Figure 1-34. Les 18 erreurs de classification commises par séparation linéaire des classes deux à deux. Pour chaque chiffre manuscrit, l'indication en haut à droite est la classe d'appartenance du chiffre indiquée dans la base, et le chiffre en bas à droite est la classe affectée par le classifieur.

**Un échec : QSAR**

- Quantitative Structure Activity Relations

Prédire certaines propriétés de molécules (par exemple activité biologique) à partir de descriptions :

- chimiques
- géométriques
- électriques

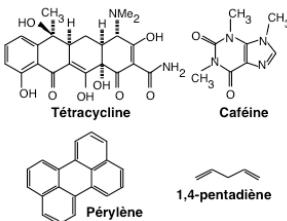


Figure 1-41. Molécules présentant des particularités chimiques dont les propriétés sont mal prédites par des réseaux de neurones.

**Rôle de la couche cachée****La base de données**

65473      60198      68544  
70065      70117      19032      98720  
27260      61828      19859  
74136      19137      63101  
20878      60521      38002  
48640-2398      20907      14868

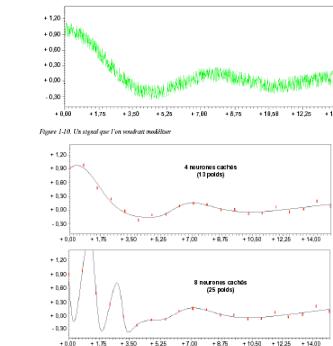
**La régression**

Figure 1-43. Plus de neurones équivaut par ailleurs, le nombre de neurones le plus important lorsque possible la meilleure propriété de généralisation.

**Rôle de la couche cachée**

## PMC : Analyse

- Rôle des cellules cachées
- Efficacité calculatoire

## PMC : La rétro-propagation de gradient (variantes)

- Ajout d'un moment

$$\Delta w_{ji}(t+1) = -\eta \frac{\partial E}{\partial w_{ji}} + \alpha \Delta w_{ji}(t)$$

## PMC : Mise en pratique (2)

### • Problèmes méthodologiques

- ❑ Choix de la fonction de coût (mesure du risque)
- ❑ Quand arrêter l'apprentissage ?
- ❑ Validation des résultats ?
- ❑ Optimisation de la capacité en généralisation ?
- ❑ Choix de la structure : nombre de couches, des cellules Dimensionnement automatique ?
- ❑ Qualité de la base d'apprentissage ?

## Généralisation : optimiser la structure d'un réseau

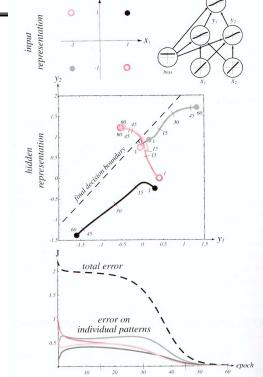
### • Par croissance progressive

- ❑ Cascade correlation [Fahlman,1990]

### • Par élagage

- ❑ Optimal brain damage [Le Cun,1990]
- ❑ Optimal brain surgeon [Hassibi,1993]

## Rôle de la couche cachée



## Sources documentaires

### Ouvrages / articles

- Dreyfus et al (2001) : Réseaux de neurones. Méthodologie et applications. Eyrolles, 2001.
- Bishop C. (95) : Neural networks for pattern recognition. Clarendon Press - Oxford, 1995.
- Haykin (98) : Neural Networks. Prentice Hall, 1998.
- Hertz, Krogh & Palmer (91) : Introduction to the theory of neural computation. Addison Wesley, 1991.
- Thiria, Gascuel, Lechevallier & Canu (97) : Statistiques et méthodes neuronales. Dunod, 1997.
- Vapnik (95) : The nature of statistical learning. Springer Verlag, 1995.

### Sites web

- <http://www.lps.ens.fr/~adal/> (point d'entrée pour de nombreux sites)

## PMC à fonctions radiales (RBF)

### Définition

- Couche cachée de cellules à fonction d'activation radiale (e.g. gaussienne)
  - Idée : "paver" l'espace des entrées avec ces "champs récepteurs"
- Couche de sortie : combinaison linéaire sur la couche cachée

### Propriétés

- Approximateur universel ([Hartman et al.,90], ...)
- Mais non parcimonieux (explosion combinatoire avec la taille des entrées)
- Réservé aux problèmes de faible dimensionnalité
- Liens étroits avec les systèmes d'inférence floue et les réseaux neuro-flous

## Les réseaux récurrents

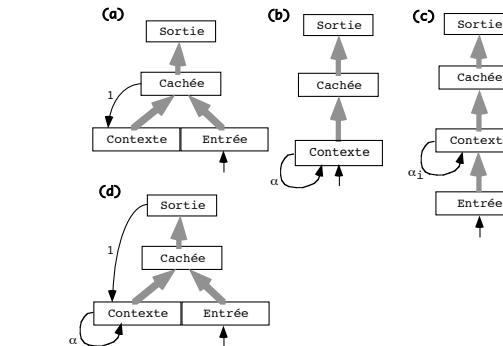
### Tâches

- Reconnaissance de séquence
  - E.g. reconnaître le mot correspondant à un signal vocal
- Reproduction de séquence
  - E.g. poursuivre la séquence quand une séquence initiale a été fournie (ex: prévision de consommation d'électricité)
- Association temporelle
  - Production d'une séquence en réponse à la reconnaissance d'une autre séquence.

### Time Delay Neural Networks (TDNNs)

- Duplication des couches (artifice : pas vraiment récurrents)

### Réseaux récurrents



## PMC à fonctions radiales (RBF) : apprentissage

### Paramètres à régler :

- Nb de cellules cachées
- Position des centres des champs récepteurs
- Diamètre des champs récepteurs
- Poids vers la couche de sortie (moyenne pondérée)

### Méthodes

- Adaptation de la rétro-propagation (possible)
- Détermination de chaque type de paramètres par une méthode propre (souvent plus efficace)
  - Centres déterminés par méthodes de "clustering" (k-means, ...)
  - Diamètres déterminés par optimisation des taux de recouvrement (PPV, ...)
  - Poids par technique d'optimisation linéaire (calcul de pseudo-inverse, ...)