

Connexionnisme (dans l'épistémologie des Sciences Cognitives)

3 questions récurrentes

1. **Quoi** ?
2. **Pourquoi** ?
3. **Comment** ?

Quoi est-ce ? Définition, illustration ?

Pourquoi c'est important, intéressant, nécessaire ? D'où ça vient, pourquoi on y a pensé, à quoi ça sert ?

Comment ça fonctionne ? Les composants, les règles de fonctionnement, les applications et leurs limites ?

[paumenu](#)

Connexionnisme et Modélisation Cognitive. Responsable : Hélène PAUGAM-MOISY, Professeur, Université Lumière Lyon 2 [hpaugam@isc ...](mailto:hpaugam@isc...)

www.isc.cnrs.fr/pau/paumenu.htm - 10k - [En cache](#) - [Pages similaires](#)

[connexionnisme ordinateur d'occasion, ordinateur portable d' ...](#)

connexionnisme, nm. [INTART] Discipline concernant les techniques de simulation des processus intelligents par des réseaux de neurones ...

www.pckado.com/e-marketing/c/connexionnisme.html - 4k - [En cache](#) - [Pages similaires](#)

[Connexionnisme sur le web](#)

Dernière mise à jour : 15/01/2002. Le connexionnisme sur le web. Nous donnons ici quelques liens vers des serveurs d'informations ...

www.supelec-rennes.fr/acth/net/net.html - 4k - [En cache](#) - [Pages similaires](#)

[Conférences dans le domaine du connexionnisme](#)

Dernière mise à jour : 31/10/2002. Conférences dans le domaine du **connexionnisme**. A venir. 21-22 novembre 2002, journées ACSEG ...

www.supelec-rennes.fr/acth/net/conferences.html - 18k - [En cache](#) - [Pages similaires](#)

[[Autres résultats, domaine www.supelec-rennes.fr](#)]

[Introduction au connexionnisme](#)

... vendre 150 ... Évaluez-le ! (201 votes), Tout public, Scientifique, Technique, 29.07.2002, Jérôme. Introduction au **connexionnisme**. Il s ...

www.vieartificielle.com/article/index.php?action=article&id=158 - 34k - [En cache](#) - [Pages similaires](#)

[\[JargonF\] connexionnisme](#)

connexionnisme. nm. [intelligence artificielle] Discipline concernant les techniques de simulation des processus intelligents par ...

www.linux-france.org/prj/jargonf/C/connexionnisme.html - 3k - [En cache](#) - [Pages similaires](#)

[Les réseaux de neurones formels et le connexionnisme](#)

... de physiologie. Les réseaux de neurones formels et le **connexionnisme**. La question fondamentale du **connexionnisme** est : comment rendre ...

www.grappa.univ-lille3.fr/~gilleron/PolyApp/node18.html - 5k - [En cache](#) - [Pages similaires](#)

[Hébergement de site internet, hébergeur de site web, Hosting ...](#)

... DEFINITION: **connexionnisme**. nm [INTART] Discipline concernant les techniques de simulation des processus intelligents par des réseaux ...

www.hebergeur.ch/support/jargon_article.php?iCodeArticle=12419 - 21k - [En cache](#) - [Pages similaires](#)

[connexionnisme - Définition - Tout-Savoir.Net](#)

... L exique. DEFINITION. **connexionnisme**. nm [INTART] Discipline concernant les techniques de simulation des processus intelligents par ...

.....



Les réseaux de neurones formels et le connexionnisme

La question fondamentale du *connexionnisme* est :

comment rendre compte des processus cognitifs à partir d'un ensemble d'unités, dotées chacune d'une faible puissance de calcul et interconnectées en réseau ?

La définition de *réseaux de neurones formels* et l'expérimentation menée sur ces réseaux permettent d'étudier et de tester cette hypothèse.

Quelques étapes dans la formalisation des réseaux de neurones :

- Première définition d'un neurone formel par McCulloch et Pitts en 1943
- Les percepts ou concepts sont physiquement représentés dans le cerveau par l'entrée en activité (simultanée) d'une *assemblée de neurones* (Donald Hebb, 1949). L'hypothèse concurrente est la spécialisation de certains neurones dans des tâches cognitives complexes (cf le fameux neurone "grand-mère").
- deux neurones entrant en activité simultanément vont être associés (c'est-à-dire que leur contacts synaptiques vont être renforcés). On parle de *loi de Hebb* et d'*associationnisme*
- Le *perceptron* de Frank Rosenblatt (1958) : le premier modèle pour lequel un processus d'*apprentissage* a pu être défini.
- Le livre de Minski et Papert "Perceptrons" (1969). Cet ouvrage contient une étude critique très complète des perceptrons. On lui reproche parfois violemment d'avoir sonné le glas des recherches sur les réseaux neuronaux dans les années 70, ce que nient leurs auteurs. Ce livre a été réédité en 1980, avec des ajouts et corrections manuscrites dans les marges, sans doute pour qu'on ne puisse pas les accuser de camoufler la première version du texte !
- l'algorithme de rétropropagation du gradient dans les réseaux multi-couches découvert au début des années 80
- le modèle de Hopfield (1982) : mémoire associative et attracteurs.
- la machine de Boltzman (1985)
- les réseaux auto-constructifs

Nous n'étudierons dans ce cours qu'une version simplifiée du Perceptron, ancêtre des réseaux de neurones formels et brique de base des modèles plus complexes, et l'algorithme de rétropropagation du gradient appliqué aux réseaux multicouches, le premier modèle vraiment convaincant.

<http://www.tout-savoir.net/lexique.php?rub=definition&code=1761>

DEFINITION

connexionnisme

n. m.

[INTART] Discipline concernant les techniques de [simulation](#) des [processus](#) intelligents par des réseaux de neurones et des ordinateurs Neuronaux. Voir [neuronal](#).

Articles voisins :

Connection [Machine](#) - [connectique](#) - [connectivité](#) - [connerietif](#) - [connexion](#) - [connexité](#)
- [CONS](#) - conseils d'utilisation

<http://www.google.fr/search?q=cache:rtaAnPgj5MwJ:www.aideeleves.net/lectures/gombert.rtf+connexionisme&hl=fr>

Ceci est la version HTML du fichier <http://www.aideeleves.net/lectures/gombert.rtf>.

Lorsque **G o o g l e** explore le Web, il crée automatiquement une version HTML des documents récupérés.

Pour créer un lien avec cette page ou l'inclure dans vos favoris/signets, utilisez l'adresse suivante :

<http://www.google.com/search?q=cache:rtaAnPgj5MwJ:www.aideeleves.net/lectures/gombert.rtf+connexionisme&hl=fr>.

Google n'est ni affilié aux auteurs de cette page ni responsable de son contenu.

Les termes de recherche suivants ont été mis en valeur :

connexionisme

Jean-Pierre Chevalier Le 9:41

L'apprentissage de la lecture: un double processus d'apprentissage pour maîtriser un double code

Conférence de Monsieur Jean-Emile GOMBERT,

Professeur de Psychologie Cognitive

Université Rennes 2

au site des Deux-Sèvres de l'IUFM de Poitou-Charentes, à Niort.

En préambule à sa conférence, Monsieur Gombert précise que l'objet de son intervention porte sur les problèmes d'acquisition des codes dans la lecture, ce qui ne représente qu'un des aspects de l'apprentissage de la lecture, par rapport à d'autres aspects tels que la prise en compte des structures textuelles, etc.....

Dans un premier temps, à partir d'un travail effectué pour le compte du Ministère de l'Education Nationale, dans le cadre des évaluations d'entrée en 6ème, le conférencier au chapeau au large bord nous aida à comprendre quels sont les processus à l'oeuvre dans la reconnaissance des mots écrits.

Processus à l'oeuvre dans la reconnaissance des mots écrits.

Les épreuves ont porté sur un échantillon de 3.000 collégiens, représentatifs de la population, en tenant compte des deux indicateurs que sont l'exactitude de la réponse et de la vitesse à répondre en un temps donné.

Les épreuves portaient sur les points suivants:

- barrer des ronds, pour évaluer la précision graphique des jeunes observés;
- détecter des homophones dans une liste, avec des mots inventés, pour évaluer la capacité de correspondance graphie-phonie;
- entourer l'écriture si elle correspond à l'image, avec des difficultés de lecture d'ordre
- sémantique (le mot ne correspond pas au dessin; le mot vache pour le dessin d'un mouton);
- orthographique du point de vue du rapport grapho-phonologique (tambron pour tambour)
- orthographique du point de vue lexical (paile pour pelle).
- détection lexicale dans une liste (barrer les mots qui n'existent pas);
- entourer l'image qui correspond à la phrase;
- barrer les mots qui ne sont pas de la même famille que le mot en gras, et montrer ainsi ses capacités à procéder à une analyse morphologique;
- segmenter un texte d'auteur non segmenté et procéder ainsi à une analyse syntaxique.

Sur les 3.000 enfants observés, 14,9 pour cent ne possèdent pas les compétences de base en français, ce qui est proche des 15 pour cent connus sur la population nationale.

Sur ces 14,9 pour cent, les épreuves complémentaires des batteries de Monsieur Gombert permettent de dire:

- rien du tout pour 5 pour cent des enfants observés;
- que 2,8 pour cent des enfants ont des difficultés du point de vue de la compréhension, qui est un processus de haut niveau;
- 2,2 pour cent des enfants sont en échec dans toutes les épreuves et présentent donc de grosses difficultés;
- 10 pour cent ont des problèmes dans les tâches d'identification des mots écrits:
 - 2,1 pour cent sont en échec du point de vue des traitements phonologiques;
 - 7,8 pour cent réussissent les exercices, mais trop lentement; ils présentent des défauts d'automatisme dans la lecture, l'analyse. Ainsi, deux tiers des enfants en difficulté en 6ème ont des problèmes dans l'automatisation du traitement de l'écrit. Ce constat pousse donc le conférencier à avancer plus loin sur sa recherche.

Aspects importants de la lecture des mots écrits.

- identifier les lettres;

- être capable de faire correspondre des lettres à des sons, et donc de maîtriser le code alphabophonétique;
- intégrer également le code grapho-morphologique des mots, c'est à dire les indices indiqués plus haut qui permettent de dépasser la phonologie pour écrire.
- le patrimoine lexical de l'enfant, le nombre de mots inconnus sémantiquement pour lui dans le texte.
- la syntaxe, du point de vue de l'ordre des mots, ainsi que de la morpho-syntaxe (le marquage de la grammaire dans les mots, tel que les accords en genre, nombre, etc....)

Monsieur Gombert en déduit donc que, face à un mot, l'individu procède par deux voies d'accès à son lexique mental

- il effectue un traitement orthographique, en lecture globale; l'identification des lettres du mot met en action un calcul orthographique;
- il effectue une analyse phonologique, après conversion grapho-phonétique, et mise en action d'un calcul phonologique.

Cette explication n'est que modèle théorique, piste à parfaire, incomplète. Elle rencontre entre autres deux limites:

- le voisinage orthographique de mots et les interférences que cela provoque alors entre les deux voies d'accès à l'identification d'un mot;
- le modèle connexionniste qui stipule (si j'ai bien compris, mais je n'en suis pas sûr) qu'il peut y avoir renforcement ou association des associations entre les deux voies, par la signification donnée aux mots dans un contexte précis: les processus orthographiques (de prise en compte des stimuli visuels que sont les écrits) se trouvent d'autant plus associés aux processus phonologiques, que les processus sémantiques et contextuels sont importants.

Importance du voisinage orthographique dans l'apprentissage de la lecture

Selon Monsieur Gombert, le voisinage orthographique et la morphologie des mots ont une place insuffisamment prise en compte par divers modèles de l'apprentissage de la lecture et de l'écriture, entre autres le modèle qui établit des stades de développement de cet apprentissage (stade logographique de reconnaissance de logos, stade alphabétique qui utilise la médiation phonologique pour la lecture des mots), et stade orthographique. Le voisinage orthographique apporterait une facilitation des l'identification des mots chez l'apprenti lecteur. Ceci fut entre autres vérifié lors de tests qui montrèrent la diminution de l'erreur de lecture pour des non-mots qui ont des voisins orthographiques.

En conséquence, Monsieur Gombert estime indispensable que les enfants apprennent à utiliser la morphologie des mots, lors de leurs apprentissages fondamentaux.

L'implicite et l'explicite dans l'apprentissage de la lecture

Pour conclure, Monsieur Gombert nous invite à comprendre que l'apprentissage de la lecture est plus le fait d'apprentissages implicites que de l'enseignement, ce qui n'est pas sans conséquence sur l'acte pédagogique. Il en serait d'ailleurs de même pour la correction orthographique.

Ces apprentissages seraient donc le fait de l'interaction

- d'apprentissages implicites qui dépendent des facteurs fréquentiels;
- de l'enseignement qui vise l'installation de connaissances explicites, contrôlées; une telle fréquentation explicite de l'écrit accélérera les apprentissages premiers.

En conséquence, le rôle de l'enseignant devient d'installer des instances de contrôle pour permettre de corriger les automatismes inappropriés. Si les connaissances explicites s'oublient, les connaissances implicites résistent à l'oubli. L'apprentissage est donc une perpétuelle interaction entre l'explicite et l'implicite chez l'individu.

Au sujet du connexionnisme:

Je n'avais pas très bien compris, et Denis Alamargot, Co-listier, formateur AIS à l'IUFM de Poitiers, et Maître de Conférence en Psychologie Cognitive a apporté les précisions suivantes:

Bonjour à toutes et tous,

Il semble que la notion de connexionnisme ne soit pas sans poser quelques problèmes conceptuels...Alors quelques mots d'explicitation:

Le connexionnisme est un modèle de traitement de l'information inventé globalement dans les années 1980 par Rumelhart et McClelland, plus exactement:

Rumelhart, D. E., & McClelland, J. L. (1986). Parallel distributed processing : explorations in the microstructure of cognition. Cambridge, M.A.: Bradford.

Le principe en est à la fois simple et compliqué. Les modèles connexionnistes s'opposent à l'approche classique des modèles symboliques.

Le modèle de Piaget est un exemple entre autres et pour aller vite, de modèles symboliques; le modèle de mémoire (MCT, MLT) de Shiffrin et Atkinson en est un autre: il existe des représentations et des

connaissances d'une part, et des processus ou des opérations mentales, d'autre part. Le traitement de l'information consiste à transformer les connaissances en d'autres connaissances via des processus. (Nous avons largement vu ça en cours de psychologie cognitive). Enfin, dans cette approche, les contraintes de la MT sont telles que les traitements sont le plus souvent envisagés comme séquentiels, car l'on ne peut pas faire plusieurs choses à la fois (sauf automatisation de certains traitements).

Le connexionnisme prend le contre-pied de l'approche symbolique en ce sens que les connaissances n'existent plus en tant que telles (plus de concepts, d'images mentales ou autre forme de connaissances), que les processus n'existent plus non plus et, troisième point, que les traitements seraient effectués en parallèle et non plus en séquentiel.

En fait, le système cognitif serait composé essentiellement d'un réseau non-symbolique de neurones artificiels (car c'est surtout un modèle testé et utilisé en intelligence artificielle) qui, comme des vrais neurones, entretiennent entre eux des rapports d'activation et d'inhibition. Lorsque des informations arrivent dans le réseau (comme dans le cas de la lecture), les potentiels d'activations et d'inhibition de tout le réseau sont changés mais ils sont changés en même temps sur la base de toutes les informations. C'est une donc configuration entière qui évolue. La connaissance est alors l'état d'une configuration à un moment donné. Si l'on poursuit les stimulations, alors le réseau va encore évoluer, faisant évoluer encore la connaissance.

Ce modèle est très pertinent dans le cas de la lecture car il permet de traiter en parallèle les différents indices issues de la trace écrite: par exemple (caricatural): un lecteur peut appréhender des indices du type:

50%graphémique+40%phonologique+10%orthographique, et faire reposer son décodage sur le traitement parallèle et pondéré de ces trois paramètres.

Puis, en lisant une page entière, par exemple, ces pondérations vont évoluer implicitement (d'où le terme "apprentissage implicite" - à ne pas confondre avec "imprégnation" qui est un concept beaucoup plus social et flou) et faire que le lecteur va, par exemple, mettre peu à peu et de plus en plus fortement, l'accent sur les indices orthographiques:

50%graphémique+10%phonologique+40%orthographique.

En fait, le connexionnisme permet de traiter plusieurs informations en même temps, de façon implicite. L'état des connaissances évolue au fur et à mesure de l'activité, de façon continue car c'est une configuration de réseau qui évolue.

Cf. le livre simple et clair, déjà conseillé: Le cerveau et la pensée, Editions Sciences Humaines, chercher le mot clé: connexionnisme.

Attention, **modèles connexionnisme et symbolique ne sont pas opposés en réalité**. Le premier permet d'expliquer la multidétermination du décodage en lecture (Cf. la conférence de Gombert), le second permet d'expliquer les phénomènes de compréhension après le décodage. En gros: connexionnisme pour les bas niveaux de traitements (graphèmes, phonèmes, orthographe), modèles symboliques pour les hauts niveaux de traitements (planification, inférences, compréhension, etc.).

Le point important, dans les études sur la lecture, est de ne pas tomber dans la religion et l'intégrisme (30 ans de débat stérile sur "analytique vs global" alors qu'on sait depuis 20 ans, en psycholinguistique, que lire est une **interaction permanente entre hauts niveaux et bas niveaux**). Lire est une activité tellement complexe et peu connue qu'il vaut mieux en réalité utiliser toutes les méthodes et toutes les théories en tant que praticien...Il serait ridicule de se priver d'une méthode de remédiation par conviction...Imaginez que votre médecin fasse de même: pas d'antibio, pas de chimio à cause de ses propres convictions ... (qui ne sont pas forcément les vôtres...).

En espérant que ces quelques lignes éclaircissent un peu les choses...

A bientôt,

Denis Alamargot

Résultat de la 1^{ère} itération

QUOI :

Définition du connexionnisme :

- rendre compte des processus cognitifs à partir d'un ensemble d'unités, dotées chacune d'une faible puissance de calcul et interconnectées en réseau
 - Discipline concernant les techniques de simulation des processus intelligents par des réseaux de neurones et des ordinateurs Neuronaux
 - Le connexionnisme est un modèle de traitement de l'information
- loi de Hebb
apprentissage

Questions :

- 1.1 réseaux de neurones
- 1.2 ordinateurs Neuronaux
- 1.3 loi de Hebb
- 1.4 apprentissage

POURQUOI :

Première définition d'un neurone formel par McCulloch et Pitts en 1943
assemblée de neurones (Donald Hebb, 1949)
perceptron de Frank Rosenblatt (1958)
inventé globalement dans les années 1980 par Rumelhart et McClelland

Questions :

- 2.1 neurone formel
- 2.2 perceptron
- 2.3 McCulloch et Pitts
- 2.4 Donald Hebb
- 2.5 Frank Rosenblatt
- 2.6 Rumelhart et McClelland

COMMENT :

s'opposent à l'approche classique des modèles symboliques
Le connexionnisme prend le contre-pied de l'approche symbolique en ce sens que les connaissances n'existent plus en tant que telles (plus de concepts, d'images mentales ou autre forme de connaissances), que les processus n'existent plus non plus et, troisième point, que les traitements seraient effectués en parallèle et non plus en séquentiel.
Modèles connexionnisme et symbolique ne sont pas opposés en réalité
interaction permanente entre hauts niveaux et bas niveaux

Questions :

- 3.1 Modèles ou approches symboliques

Apprentissage automatique : les réseaux de neurones

Précédent Index Suivant Chapitre 3, Apprentissage automatique : les réseaux de neurones. ...

3.1.2, Le connexionnisme et les réseaux de neurones formels. ...

www.grappa.univ-lille3.fr/polys/apprentissage/sortie005.html - 101k - [En cache](#) - [Pages similaires](#)

Comment l'homme fait-il pour raisonner, parler, calculer, apprendre, ...? Comment s'y prendre pour créer une ou de l'intelligence artificielle ? Deux types d'approches ont été essentiellement explorées :

- procéder d'abord à l'analyse logique des tâches relevant de la cognition humaine et tenter de les reconstituer par programme. C'est cette approche qui a été privilégiée par l'Intelligence Artificielle et la psychologie cognitive classiques. Cette démarche est étiquetée sous le nom de *cognitivisme*.
- puisque la pensée est produite par le cerveau ou en est une propriété, commencer par étudier comment celui-ci fonctionne. C'est cette approche qui a conduit à l'étude de réseaux de neurones formels. On désigne par *connexionnisme* la démarche consistant à vouloir rendre compte de la cognition humaine par des réseaux de neurones.

La seconde approche a donc menée à la définition et l'étude de réseaux de neurones formels qui sont des réseaux complexes d'unités de calcul élémentaire interconnectées. Il existe deux courants de recherche sur les réseaux de neurones : un premier motivé par l'étude et la modélisation des phénomènes naturels d'apprentissage à l'aide de réseaux de neurones, la pertinence biologique est importante ; un second motivé par l'obtention d'algorithmes efficaces ne se préoccupant pas de la pertinence biologique. Nous nous plaçons du point de vue du second groupe. En effet, bien que les réseaux de neurones formels aient été définis à partir de considérations biologiques, pour la plupart d'entre eux, et en particulier ceux étudiés dans ce cours, de nombreuses caractéristiques biologiques (le temps, la mémoire, ...) ne sont pas prises en compte. Toutefois, nous donnons, dans la suite de cette introduction, un bref aperçu de quelques propriétés élémentaires de neurophysiologie qui permettent au lecteur de relier neurones réels et neurones formels. Nous donnons ensuite un rapide historique des réseaux de neurones. Enfin, nous donnons une classification des différents types de réseau et les principales applications.

...

Un réseau de neurones formels est constitué d'un grand nombre de cellules de base interconnectées. De nombreuses variantes sont définies selon le choix de la cellule élémentaire, de l'architecture du réseau et de la dynamique du réseau.

Une cellule élémentaire peut manipuler des valeurs binaires ou réelles. Les valeurs binaires sont représentées par 0 et 1 ou -1 et 1. Différentes fonctions peuvent être utilisées pour le calcul de la sortie. Le calcul de la sortie peut être déterministe ou probabiliste.

L'architecture du réseau peut être sans rétroaction, c'est à dire que la sortie d'une cellule ne peut influencer son entrée. Elle peut être avec rétroaction totale ou partielle.

La dynamique du réseau peut être synchrone : toutes les cellules calculent leurs sorties respectives simultanément. La dynamique peut être asynchrone. Dans ce dernier cas, on peut avoir une dynamique asynchrone séquentielle : les cellules calculent leurs sorties chacune à son tour en séquence ou avoir une dynamique asynchrone aléatoire.

Par exemple, si on considère des neurones à sortie stochastique -1 ou 1 calculée par une fonction à seuil basée sur la fonction sigmoïde, une interconnection complète et une dynamique synchrone, on obtient le modèle de Hopfield et la notion de mémoire associative.

Si on considère des neurones déterministes à sortie réelle calculée à l'aide de la fonction sigmoïde, une architecture sans rétroaction en couches successives avec une couche d'entrées et une couche de sorties, une dynamique asynchrone séquentielle, on obtient le modèle du Perceptron multi-couches (PMC) qui sera étudié dans les paragraphes suivants.

Applications des réseaux de neurones

Les principales applications des réseaux de neurones sont l'optimisation et l'apprentissage. En apprentissage, les réseaux de neurones sont essentiellement utilisés pour :

- l'apprentissage supervisé ;
- l'apprentissage non supervisé ;

- l'apprentissage par renforcement.

Pour ces trois types d'apprentissage, il y a également un choix traditionnel entre :

- l'apprentissage << off-line >> : toutes les données sont dans une base d'exemples d'apprentissage qui sont traités simultanément ;
- l'apprentissage << on-line >> : Les exemples sont présentés les uns après les autres au fur et à mesure de leur disponibilité

Nous nous limitons, dans ce cours, à l'apprentissage supervisé à partir d'une base d'exemples. Dans ce cadre, l'apprentissage à l'aide de réseaux de neurones est bien adapté pour l'apprentissage à partir de données complexes (images sur une rétine, sons, ...) mais aussi à partir de données symboliques. Les entrées peuvent être représentées par de nombreux attributs à valeurs réelles ou symboliques, les attributs pouvant être dépendants ou non. La ou les sorties peuvent être réelles ou discrètes. L'apprentissage à l'aide de réseaux de neurones est tolérant au bruit et aux erreurs. Le temps d'apprentissage peut être long, par contre, après apprentissage, le calcul des sorties à partir d'un vecteur d'entrée est rapide. La critique principale est que le résultat de l'apprentissage, c'est-à-dire le réseau de neurones calculé par l'algorithme d'apprentissage, n'est pas interprétable par l'utilisateur : on ne peut pas donner d'explication au calcul d'une sortie sur un vecteur d'entrée. On parle de << boîte noire >>. Ceci est la principale différence entre réseaux de neurones et arbres de décision. Si l'utilisateur a besoin de pouvoir interpréter le résultat de l'apprentissage, il choisira un système basé sur les arbres de décision, sinon les deux méthodes sont concurrentes.

Nous n'étudions que le perceptron, brique de base des modèles plus complexes, et le perceptron multi-couches (PMC). L'accent sera mis sur les algorithmes d'apprentissage pour ces deux modèles, en particulier sur l'algorithme de rétropropagation du gradient appliqué aux PMC. Cet algorithme est, en effet, le premier algorithme d'apprentissage convaincant dans un modèle suffisamment puissant et cet algorithme a de nombreuses applications.

Le Perceptron

Le *perceptron* est un modèle de réseau de neurones avec algorithme d'apprentissage créé par Frank Rosenblatt en 1958. La version ci-dessous est simplifiée par rapport à l'originale. Vous trouverez une description de cette dernière dans l'exercice [??](#).

Définition 5 Un perceptron linéaire à seuil (voir figure [??](#)) prend en entrée n valeurs x_1, \dots, x_n et calcule une sortie o . Un perceptron est défini par la donnée de $n+1$ constantes : les coefficients synaptiques w_1, \dots, w_n et le seuil (ou le biais) θ . La sortie o est calculée par la formule :

Figure 3.2 : Le perceptron avec seuil

Les entrées x_1, \dots, x_n peuvent être à valeurs dans $\{0,1\}$ ou réelles, les poids peuvent être entiers ou réels. Une variante très utilisée de ce modèle est de considérer une fonction de sortie prenant ses valeurs dans $\{-1,1\}$ plutôt que dans $\{0,1\}$. Il existe également des modèles pour lesquels le calcul de la sortie est probabiliste. Dans la suite de cette partie sur le perceptron, nous considérerons toujours le modèle déterministe avec une sortie calculée dans $\{0,1\}$.

Pour simplifier les notations et certaines preuves, nous allons remplacer le seuil par une entrée supplémentaire x_0 qui prend toujours comme valeur d'entrée la valeur $x_0=1$. À cette entrée est associée un coefficient synaptique w_0 . Le modèle correspondant est décrit dans la figure [??](#). On peut décomposer le calcul de la sortie o en un premier calcul de la quantité $\sum_i w_i x_i$ appelée *potentiel post-synaptique* ou l'*entrée totale* suivi d'une application d'une *fonction d'activation* sur cette entrée totale. La fonction d'activation est la fonction de Heaviside définie par :

Figure 3.3 : Le perceptron avec entrée supplémentaire

Bien que considérant une entrée supplémentaire x_0 , un perceptron est toujours considéré comme associant une sortie o aux n entrées x_1, \dots, x_n . L'équivalence entre le modèle avec seuil et le modèle avec entrée supplémentaire à 1 est immédiate : le coefficient w_0 est l'opposé du seuil θ . Nous considérerons toujours ce dernier modèle de perceptron linéaire à seuil par la suite.

Pour passer du modèle avec sorties à valeurs dans $\{0,1\}$ au modèle à valeurs dans $\{-1,1\}$, il suffit de remplacer la fonction de Heaviside f par la fonction g définie par : $g(x) = 2f(x) - 1$. D'autres fonctions d'activation peuvent également être utilisées.

Exemple 10 *Un perceptron qui calcule le OU logique avec les deux versions : seuil ou entrée supplémentaire est présenté dans la figure ??.*

Figure 3.4 : perceptrons qui calculent le OU

On voit que quelques uns des traits principaux des neurones réels ont été retenus dans la définition du perceptron : les entrées modélisent les dendrites, les impulsions en entrée sont pondérées par les coefficients synaptiques et l'impulsion émise, c'est-à-dire la sortie, obéit à un effet de seuil (pas d'impulsion si l'entrée totale est trop faible).

Un perceptron à n entrées réelles (respectivement binaires) est une fonction de R^n (respectivement $\{0,1\}^n$) dans $\{0,1\}$. Si l'on veut faire le lien avec les chapitres précédents, on peut voir les neurones d'entrées comme décrivant un espace de description avec des attributs réels (respectivement binaires) et le perceptron comme une procédure de classification binaire (c'est-à-dire en deux classes) sur cet espace. Un système d'apprentissage à base de perceptrons doit générer, à partir d'un ensemble d'apprentissage, une hypothèse qui est un perceptron. Nous nous intéressons, dans la section suivante, à cet espace d'hypothèses, c'est-à-dire à l'étude des fonctions calculables par perceptron.

Interprétation géométrique et limitations

Définition 6 Soit S un ensemble d'exemples dans $R^n \times \{0,1\}$. On note $S_0 = \{s \in R^n \mid (s,0) \in S\}$ et $S_1 = \{s \in R^n \mid (s,1) \in S\}$. On dit que S est linéairement séparable s'il existe un hyperplan H de R^n tel que les ensembles S_0 et S_1 soient situés de part et d'autre de cet hyperplan.

Théorème 2 Un perceptron linéaire à seuil à n entrées divise l'espace des entrées R^n en deux sous-espaces délimités par un hyperplan. Réciproquement, tout ensemble linéairement séparable peut être discriminé par un perceptron.

Démonstration : Il suffit pour s'en convaincre de se rappeler que l'équation d'un hyperplan dans un espace de dimension n est de la forme :

$$\alpha_1 x_1 + \dots + \alpha_n x_n = \beta$$

Un perceptron est donc un discriminant linéaire. On montre facilement qu'un échantillon de R^n est séparable par un hyperplan si et seulement si l'échantillon de R^{n+1} obtenu en rajoutant une entrée toujours égale à 1 est séparable par un hyperplan passant par l'origine.

Toute fonction de R^n dans $\{0,1\}$ est-elle calculable par perceptron ? La réponse est évidemment non. De même, toute fonction booléenne peut-elle être calculée par un perceptron ? La réponse est également non. Le contre-exemple le plus simple est le << OU exclusif >> (XOR) sur deux variables.

Théorème 3 *Le XOR ne peut pas être calculé par un perceptron linéaire à seuil.*

Démonstration :

Démonstration algébrique : Supposons qu'il existe un perceptron défini par les coefficients synaptiques (w_0, w_1, w_2) calculant le XOR sur deux entrées booléennes x_1 et x_2 . On devrait avoir :

$$w_0 + 0 w_1 + 0 w_2 = w_0 \leq 0 \quad (3.1)$$

$$w_0 + 0 w_1 + 1 w_2 = w_0 + w_2 > 0 \quad (3.2)$$

$$w_0 + 1 w_1 + 0 w_2 = w_0 + w_1 > 0 \quad (3.3)$$

$$w_0 + 1 w_1 + 1 w_2 = w_0 + w_1 + w_2 \leq 0 \quad (3.4)$$

Il suffit d'additionner l'équation ?? et l'équation ?? d'une part, l'équation ?? et l'équation ?? d'autre part pour se rendre compte que l'hypothèse est absurde.

Démonstration géométrique : on << voit >> bien qu'aucune droite ne peut séparer les points de coordonnées (0,0) et (1,1) des points de coordonnées (0,1) et (1,0) (voir Figure ??). Si on considère une entrée $x_0=1$, il n'existe pas de plan passant par l'origine qui sépare les points de coordonnées (1,0,0) et (1,1,1) des points de coordonnées (1,0,1) et (1,1,0).

Figure 3.5 : Comment trouver une droite séparant les points (0,0) et (1,1) des points (0,1) et (1,0) ?

Algorithme d'apprentissage par correction d'erreur

Présentation de l'algorithme

Étant donné un échantillon d'apprentissage S de $R^n \times \{0,1\}$ (respectivement $\{0,1\}^n \times \{0,1\}$), c'est-à-dire un ensemble d'exemples dont les descriptions sont sur n attributs réels (respectivement binaires) et la classe est binaire, il s'agit de trouver un algorithme qui infère à partir de S un perceptron qui classe correctement les éléments de S au vu de leurs descriptions si c'est possible ou au mieux sinon.

Exemple 11 Pour apprendre la notion de chiffre pair ou impair, on peut considérer un échantillon composé des 10 chiffres écrits sur une rétine à 7 leds.

Figure 3.6 : Les 10 chiffres sur une rétine à 7 leds

En représentant chaque chiffre par le symbole qui le désigne habituellement, un échantillon d'apprentissage complet est :

$S = \{(1111110,0), (0110000,1), (1101101,0), (1111001,1), (0010011,0), (1011011,1), (0011111,0), (1110000,1), (1111111,0), (1111011,1)\}$. Le but sera d'inférer, à partir de S , un perceptron qui prend ses entrées dans $\{0,1\}^7$ et qui retourne la classe 0 si le vecteur d'entrée correspond à un chiffre pair et 1 sinon. Sur cet exemple, l'échantillon S est complet (toutes les entrées possibles sont décrites). Il est fréquent, pour les problèmes concrets, d'avoir un échantillon non complet.

L'algorithme d'apprentissage peut être décrit succinctement de la manière suivante. On initialise les poids du perceptron à des valeurs quelconques. A chaque fois que l'on présente un nouvel exemple, on ajuste les poids selon que le perceptron l'a correctement classé ou non. L'algorithme s'arrête lorsque tous les exemples ont été présentés sans modification d'aucun poids.

Dans la suite, nous noterons x^{\rightarrow} une description qui sera un élément de R^n ou $\{0,1\}^n$. La i -ème composante de x^{\rightarrow} sera notée x_i . Un échantillon S est un ensemble de couples (x^{\rightarrow}, c) où c est la classe de x^{\rightarrow} . Lorsqu'il sera utile de désigner un élément particulier de S , nous noterons $(x^{\rightarrow s}, c^s)$ le s -ième élément de S . x_i^s désignera la i -ème composante du vecteur d'entrée $x^{\rightarrow s}$. Si une entrée $x^{\rightarrow s}$ est présentée en entrée d'un perceptron, nous noterons o^s la sortie binaire calculée par le perceptron. Nous rappelons qu'il existe une $n+1$ -ième entrée x_0 de valeur 1 pour le perceptron.

L'algorithme d'apprentissage par correction d'erreur du perceptron linéaire à seuil est :

Algorithme par correction d'erreur:

Entrée : un échantillon S de $R^n \times \{0,1\}$ ou $\{0,1\}^n \times \{0,1\}$
 Initialisation aléatoire des poids w_i pour i entre 0 et n

Répéter

Prendre un exemple (x^{\rightarrow}, c) dans S

Calculer la sortie o du perceptron pour l'entrée x^{\rightarrow}

-- Mise à jour des poids --

Pour i de 0 à n

$$w_i \leftarrow w_i + (c-o)x_i$$

finpour

finRépéter

Sortie : Un perceptron P défini par (w_0, w_1, \dots, w_n)

La procédure d'apprentissage du perceptron est une procédure de *correction d'erreur* puisque les poids ne sont pas modifiés lorsque la sortie attendue c est égale à la sortie calculée o par le perceptron courant. Étudions les modifications sur les poids lorsque c diffère de o :

- si $o=0$ et $c=1$, cela signifie que le perceptron n'a pas assez pris en compte les neurones actifs de l'entrée (c'est-à-dire les neurones ayant une entrée à 1) ; dans ce cas, $w_i \leftarrow w_i + x_i$; l'algorithme ajoute la valeur de la rétine aux poids synaptiques (*renforcement*).
- si $o=1$ et $c=0$, alors $w_i \leftarrow w_i - x_i$; l'algorithme retranche la valeur de la rétine aux poids synaptiques (*inhibition*).

Remarquons que, en phase de calcul, les constantes du perceptron sont les poids synaptiques alors que les variables sont les entrées. Tandis que, en phase d'apprentissage, ce sont les coefficients synaptiques qui sont variables alors que les entrées de l'échantillon S apparaissent comme des constantes.

Certains éléments importants ont été laissés volontairement imprécis. En premier lieu, il faut préciser comment est fait le choix d'un élément de S : aléatoirement ? En suivant un ordre prédéfini ? Doivent-ils être tous présentés ? Le critère d'arrêt de la boucle principale de l'algorithme n'est pas défini : après un certain nombre d'étapes ? Lorsque tous les exemples ont été présentés ? Lorsque les poids ne sont plus modifiés pendant un certain nombre d'étapes ? Nous reviendrons sur toutes ces questions par la suite. Tout d'abord, examinons le comportement de l'algorithme sur deux exemples :

Exemple 12 Apprentissage du OU : les descriptions appartiennent à $\{0,1\}^2$, les entrées du perceptron appartiennent à $\{0,1\}^3$, la première composante correspond à l'entrée x_0 et vaut toujours 1, les deux composantes suivantes correspondent aux variables x_1 et x_2 . On suppose qu'à l'initialisation, les poids suivants ont été choisis : $w_0=0$; $w_1 = 1$ et $w_2 = -1$. On suppose que les exemples sont présentés dans l'ordre lexicographique.

| étape | w_0 | w_1 | w_2 | Entrée | $\sum_0^2 w_i x_i$ | o | c | w_0 | w_1 | w_2 |
|-------|-------|-------|-------|--------|--------------------|-----|-----|------------|------------|------------|
| init | | | | | | | | 0 | 1 | -1 |
| 1 | 0 | 1 | -1 | 100 | 0 | 0 | 0 | $0+0x1$ | $1+0x0$ | $-1+0x0$ |
| 2 | 0 | 1 | -1 | 101 | -1 | 0 | 1 | $0+1x1$ | $1+1x0$ | $-1+1x1$ |
| 3 | 1 | 1 | 0 | 110 | 2 | 1 | 1 | 1 | 1 | 0 |
| 4 | 1 | 1 | 0 | 111 | 2 | 1 | 1 | 1 | 1 | 0 |
| 5 | 1 | 1 | 0 | 100 | 1 | 1 | 0 | $1+(-1)x1$ | $1+(-1)x0$ | $0+(-1)x0$ |
| 6 | 0 | 1 | 0 | 101 | 0 | 0 | 1 | $0+1x1$ | $1+1x0$ | $0+1x1$ |
| 7 | 1 | 1 | 1 | 110 | 2 | 1 | 1 | 1 | 1 | 1 |
| 8 | 1 | 1 | 1 | 111 | 3 | 1 | 1 | 1 | 1 | 1 |
| 9 | 1 | 1 | 1 | 100 | 1 | 1 | 0 | $1+(-1)x1$ | $1+(-1)x0$ | $1+(-1)x0$ |
| 10 | 0 | 1 | 1 | 101 | 1 | 1 | 1 | 0 | 1 | 1 |

Aucune entrée ne modifie le perceptron à partir de cette étape. Vous pouvez aisément vérifier que ce perceptron calcule le OU logique sur les entrées x_1 et x_2 .

Exemple 13 Apprentissage d'un ensemble linéairement séparable : les descriptions appartiennent à R^2 , le concept cible est défini à l'aide de la droite d'équation $y=x/2$. Les couples (x,y) tels que $y>x/2$ sont de classe 1 ; Les couples (x,y) tels que $y \leq x/2$ sont de classe 0. L'échantillon d'entrée est $S=\{(0,2),1), ((1,1),1), ((1,2.5),1), ((2,0),0), ((3,0.5),0)\}$. On suppose qu'à l'initialisation, les poids suivants ont été choisis : $w_0=0$; $w_1 = 0$ et $w_2 = 0$. On choisit de présenter tous les exemples en alternant exemple positif (de classe 1) et exemple négatif.

| étape | w_0 | w_1 | w_2 | Entrée | $\sum_0^2 w_i x_i$ | o | c | w_0 | w_1 | w_2 |
|-------|-------|-------|-------|-----------|--------------------|-----|-----|-------|-------|-------|
| init | | | | | | | | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | (1,0,2) | 0 | 0 | 1 | 1 | 0 | 2 |
| 2 | 1 | 0 | 2 | (1,2,0) | 1 | 1 | 0 | 0 | -2 | 2 |
| 3 | 0 | -2 | 2 | (1,1,1) | 0 | 0 | 1 | 1 | -1 | 3 |
| 4 | 1 | -1 | 3 | (1,3,0.5) | -0.5 | 0 | 0 | 1 | -1 | 3 |
| 5 | 1 | -1 | 3 | (1,1,2.5) | 7.5 | 1 | 1 | 1 | -1 | 3 |

Aucune entrée ne modifie le perceptron à partir de cette étape car ce perceptron classe correctement tous les exemples de S . Le perceptron de sortie associe la classe 1 aux couples (x,y) tels que $y>x/3 - 1/3$.

Figure 3.7 : échantillon S ; hyperplans séparateurs cible et appris

Dans les deux exemples, l'échantillon d'apprentissage est un ensemble linéairement séparable. Lors de la phase d'apprentissage, tous les exemples sont présentés jusqu'à la convergence, c'est-à-dire jusqu'à ce qu'une présentation complète des exemples n'entraîne aucune modification de l'hypothèse en cours. Nous démontrons, dans la section suivante, que ceci est un résultat général.

Théorème d'apprentissage par correction d'erreur

Théorème 4 Si l'échantillon S est linéairement séparable et si les exemples sont présentés équitablement (c'est-à-dire que la procédure de choix des exemples n'en exclut aucun), la procédure d'apprentissage par correction d'erreur converge vers un perceptron linéaire à seuil qui calcule S .

Démonstration :

Soit un échantillon d'entrée sur n variables réelles (le cas de variables binaires s'en déduit), soit S l'échantillon obtenu en ajoutant une $n+1$ -ième entrée x_0 toujours égale à 1, par hypothèse l'échantillon est linéairement séparable, donc il existe un hyperplan de R^{n+1} passant par l'origine qui sépare S , soit encore, il existe un vecteur $v \rightarrow=(v_0, \dots, v_n)$ de R^{n+1} tel que :

....

Critiques sur la méthode par correction d'erreur

Nous venons de démontrer que si l'échantillon est linéairement séparable, si tous les exemples sont présentés équitablement et que le critère d'arrêt est la stabilité de l'hypothèse après une présentation complète de l'échantillon alors l'algorithme s'arrête avec un perceptron qui classe correctement l'échantillon d'apprentissage.

Que se passe-t-il si l'échantillon d'entrée n'est pas linéairement séparable ? L'inconvénient majeur de cet algorithme est que si l'échantillon présenté n'est pas linéairement séparable, l'algorithme ne convergera pas et l'on aura aucun moyen de le savoir. On pourrait penser qu'il suffit d'observer l'évolution des poids synaptiques pour en déduire si l'on doit arrêter ou non l'algorithme. En effet, si les poids et le seuil prennent deux fois les mêmes valeurs sans que le perceptron ait appris et alors que tous les exemples ont été présentés, cela signifie d'après le théorème précédent que l'échantillon n'est pas séparable. Et l'on peut penser que l'on peut borner les poids et le seuil en fonction de la taille de la rétine. C'est vrai mais les résultats de complexité énoncés ci-dessous (sans démonstration) montrent que cette idée n'est pas applicable en pratique.

Théorème 5

1. Toute fonction booléenne linéairement séparable sur n variables peut être implantée par un perceptron dont les poids synaptiques entiers w_i sont tels que $|w_i| \leq (n+1)^{n+1/2}$.
2. Il existe des fonction booléennes linéairement séparables sur n variables qui requièrent des poids entiers supérieurs à $2^{n+1/2}$.

Ces résultats sont assez décevants. Le premier montre que l'on peut borner les poids synaptiques en fonction de la taille de la rétine, mais par un nombre tellement grand que toute application pratique de ce résultat semble exclue. Le second résultat montre en particulier que l'algorithme d'apprentissage peut nécessiter un nombre exponentiel d'étapes (en fonction de la taille de la rétine) avant de s'arrêter. En effet, les poids ne varient qu'au plus d'une unité à chaque étape.

Même lorsque l'algorithme d'apprentissage du perceptron converge, rien ne garantit que la solution sera *robuste*, c'est-à-dire qu'elle ne sera pas remise en cause par la présentation d'un seul nouvel exemple. Pour s'en persuader, il suffit de se reporter à l'exemple ???. Supposons qu'on ajoute l'exemple ((3,1),0), cet exemple remet en cause l'hypothèse générée car le perceptron sorti par notre algorithme associe la classe 1 à la description (3,1). Un « bon » algorithme d'apprentissage devrait produire une solution robuste. Graphiquement, si on considère un échantillon linéairement séparable, une solution robuste serait « à mi-chemin » entre les points de classe 1 et de classe 0 comme le montre la Figure ???.

Figure 3.8 : Un nouvel exemple peut remettre en cause le perceptron appris.

Pire encore, cet algorithme n'a aucune tolérance au « bruit » : si du bruit, c'est-à-dire une information mal classée, vient perturber les données d'entrée, le perceptron ne convergera jamais. En effet, des données linéairement séparables peuvent ne plus l'être à cause du bruit. En particulier, les problèmes *non-déterministes*, c'est-à-dire pour lesquels une même description peut représenter des éléments de classes différentes ne peuvent pas être traités à l'aide d'un perceptron. Si on considère les données de l'exemple présenté dans la Figure ???, les données ne sont pas linéairement séparables, mais un « bon » algorithme d'apprentissage pour le perceptron devrait être capable de produire un séparateur linéaire comme celui qui est présenté dans cette même figure, ce qui n'est pas le cas de l'algorithme par correction d'erreur.

Figure 3.9 : Apprentissage en présence de bruit

Le but des sections suivantes est de présenter des algorithmes d'apprentissage du perceptron qui produisent des solutions robustes pour des échantillons linéairement séparables et des solutions « approximatives » pour des échantillons non linéairement séparables.

3.2.5 Conclusion

En conclusion, l'apprentissage par perceptron ou par la méthode du gradient ne sont rien d'autre que des techniques de séparation linéaire qu'il faudrait comparer aux techniques utilisées habituellement en statistiques. Ces méthodes sont non paramétriques, c'est-à-dire qu'elles n'exigent aucune autre hypothèse sur les données que la séparabilité.

On peut montrer que « presque » tous les échantillons de moins de $2n$ exemples sont linéairement séparables lorsque n est le nombre de variables. Une classification correcte d'un petit échantillon n'a donc aucune valeur prédictive. Par contre, lorsque l'on travaille sur suffisamment de données et que le problème s'y prête, on constate empiriquement que le perceptron appris par un des algorithmes précédents a un bon pouvoir prédictif.

Il est bien évident que la plupart des problèmes d'apprentissage qui se posent naturellement ne peuvent pas être résolus par des méthodes aussi simples : il n'y a que très peu d'espoir que les exemples « naturels » se répartissent « sagement » de part et d'autre d'un hyperplan. Une manière de résoudre cette difficulté serait soit de mettre au point des séparateurs non-linéaires, soit (ce qui revient à peu près au même) de complexifier

l'espace de représentation de manière à linéariser le problème initial. C'est ce que permettent de faire les réseaux multicouches que nous étudions maintenant.

Les réseaux multi-couches

Introduction et définition de l'architecture

Un perceptron linéaire à seuil est bien adapté pour des échantillons linéairement séparables. Cependant, dans la plupart des problèmes réels, cette condition n'est pas réalisée. Un perceptron linéaire à seuil est constitué d'un seul neurone. On s'est très vite rendu compte qu'en combinant plusieurs neurones le pouvoir de calcul était augmenté. Par exemple, dans le cas des fonctions booléennes, il est facile de calculer le XOR en utilisant deux neurones linéaires à seuil. Cet exemple est présenté dans la figure ??.

Figure 3.11 : Il suffit de rajouter un neurone intermédiaire entre la rétine et la cellule de décision pour pouvoir calculer le XOR

La notion de perceptron multi-couches (PMC) a ainsi été définie. On considère une couche d'entrée qui correspond aux variables d'entrée, une couche de sorties, et un certain nombre de couches intermédiaires. Les liens n'existent qu'entre les cellules d'une couche avec les cellules de la couche suivante. Le XOR peut être calculé par un perceptron multi-couches présenté dans la figure ?? en transformant légèrement le réseau présenté dans la figure ??.

Figure 3.12 : PMC pour le XOR ; les liens avec poids nul ne sont pas représentés

Définition 8 Un réseau de neurones à couches cachées est défini par une architecture vérifiant les propriétés suivantes :

- les cellules sont réparties de façon exclusive dans des couches C_0, C_1, \dots, C_q ,
- la première couche C_0 est la rétine composée des cellules d'entrée qui correspondent aux n variables d'entrée ; les couches C_1, \dots, C_{q-1} sont les couches cachées ; la couche C_q est composée de la (ou les) cellule(s) de décision,
- Les entrées d'une cellule d'une couche C_i avec $i \geq 1$ sont toutes les cellules de la couche C_{i-1} et aucune autre cellule.

La dynamique du réseau est synchrone.

Le réseau présenté dans la figure ?? pour le calcul du XOR est un réseau à une couche cachée. L'architecture d'un réseau à couches cachées est sans rétroaction. Dans notre définition, nous avons supposé qu'une cellule avait pour entrée toutes les cellules de la couche précédente, ce peut être un sous-ensemble des cellules de la couche précédente. Ce qui est primordial dans la définition, c'est que les entrées appartiennent uniquement à la couche précédente, c'est-à-dire que la structure en couches est respectée et qu'il n'y a pas de rétroaction.

Supposons que les cellules élémentaires soient des perceptrons linéaires à seuil, on parle alors de perceptrons multi-couches (PMC) linéaire à seuil. Soit n variables binaires, il est facile de montrer que le OU n -aire est calculable par un perceptron linéaire à seuil et que toute conjonction sur les littéraux définis à partir des n variables est calculable par un perceptron linéaire à seuil. Étant donné une fonction booléenne sur n variables, cette fonction peut être mise sous forme normale disjonctive, il suffit alors que chaque cellule de la couche cachée calcule une conjonction et que la cellule de sortie calcule la disjonction des résultats. Nous avons ainsi démontré que :

Proposition 1 Toute fonction booléenne peut être calculée par un PMC linéaire à seuil comprenant une seule couche cachée.

Cependant, si l'on utilise cette méthode pour construire un réseau de neurones pour calculer une fonction booléenne quelconque, la couche cachée pourra contenir jusqu'à 2^n neurones (où n est la taille de la rétine), ce qui est inacceptable en pratique. On peut montrer par ailleurs que cette solution est loin d'être la meilleure (voir le cas de la fonction parité dans l'exercice ??).

Pour pouvoir utiliser les réseaux multi-couches en apprentissage, deux choses sont indispensables :

- une méthode indiquant comment choisir une *architecture* de réseau pour résoudre un problème donné. C'est-à-dire, pouvoir répondre aux questions suivantes : combien de couches cachées ? combien de neurones par couches cachées ?
- une fois l'architecture choisie, un algorithme d'apprentissage qui calcule, à partir de l'échantillon d'apprentissage, les valeurs des coefficients synaptiques pour construire un réseau adapté au problème.

Le premier point est encore un sujet de recherche actif : quelques algorithmes d'apprentissage *auto-constructifs* ont été proposés. Leur rôle est double :

- apprentissage de l'échantillon avec un réseau courant,
- modification du réseau courant, en ajoutant de nouvelles cellules ou une nouvelle couche, en cas d'échec de l'apprentissage.

Il semble assez facile de concevoir des algorithmes auto-constructifs qui classent correctement l'échantillon, mais beaucoup plus difficile d'en obtenir qui aient un bon pouvoir de généralisation.

Il a fallu attendre le début des années 80 pour que le deuxième problème trouve une solution : *l'algorithme de rétropropagation du gradient*, découvert simultanément par des équipes française et américaine. Cet algorithme est, comme son nom l'indique, basé sur la méthode du gradient. Il est donc nécessaire de considérer des fonctions d'erreur dérivables. Ceci implique qu'il n'est pas possible de considérer comme cellule élémentaire un perceptron linéaire à seuil. L'idée est alors de prendre comme cellule élémentaire un perceptron linéaire. Malheureusement, dans ce cas, l'introduction de cellules supplémentaires n'augmente pas l'expressivité. En effet, une combinaison linéaire de fonctions linéaires est une fonction linéaire ! Nous allons donc avoir besoin de considérer une nouvelle cellule élémentaire. La sortie de cette cellule sera une fonction de variable réelle dérivable qui est une approximation de la fonction de Heaviside. Nous donnons dans la section suivante la définition d'une telle cellule et présentons l'algorithme de rétropropagation du gradient.

3.3.2 L'algorithme de rétropropagation du gradient

3.3.3 Applications

De nombreuses applications de l'algorithme de rétropropagation du gradient ont été réalisées. Parmi les plus souvent citées, nous en mentionnons deux : *NetTalk* de Sejnowski et les familles italo-américaines de Hinton

NetTalk

NetTalk est un réseau qui a appris à transformer un texte (en anglais) en une suite de phonèmes correspondant à sa lecture. Couplé en entrée à un scanner et à un OCR et en sortie à un synthétiseur de paroles, ce réseau est donc capable de lire un texte à haute voix.

Description de l'architecture du réseau :

- la couche d'entrée comprend 7 groupes de 29 neurones. Chaque groupe correspond à un caractère codé directement (non compressé). Les 7 caractères en entrée forment un contexte local de trois caractères entourant de part et d'autre un caractère central. Par exemple, le caractère 'c' se prononce différemment dans 'cygne' et dans 'carte'. Les ambiguïtés les plus courantes semblent pouvoir être levés avec un tel contexte.
- la couche cachée contient 80 neurones
- la couche de sortie comprend 26 neurones servant à coder les caractéristiques des phonèmes : la *zone vibratoire* (labiale, dentale, ...), le *type* de phonème (arrêt, nasale, fricative, ...), la *hauteur* des voyelles, la *ponctuation* (silence, pause, élision, arrêt net), l'*accentuation*, ...

Figure 3.15 : NetTalk

Le réseau comprend donc au total 309 neurones et comme les connexions sont complètes d'une couche à l'autre, 18320 connexions.

D'après les résultats publiés : 50000 mots appartenant à un corpus de 1000 mots ont été présentés au réseau. Le temps d'apprentissage a été : une nuit sur un VAX 780. Les performances : 95% pour l'ensemble d'apprentissage et 75% pour les nouveaux mots.

Citons à ce propos Jean-Pierre Nadal : << Dans ses conférences, T. Sejnowski faisait entendre à l'auditoire un enregistrement sonore pris à divers moments au cours de la phase d'apprentissage. On pouvait alors entendre le réseau d'abord balbutier, puis on distinguait un découpage du texte en phrases, jusqu'à finalement une lecture raisonnable du texte. L'effet est évidemment spectaculaire, et il n'y a pas de doute, qu'à la suite de ces démonstrations, nombreux sont ceux qui se sont convertis au connexionnisme, si je puis dire ... On a ainsi vu, et ceci principalement aux Etats-Unis, se développer la vague, née en 1985, d'une activité impressionnante de ceux pour qui, << ça y était >> : pour résoudre n'importe quel problème, il suffit de mettre dans une boîte noire quelques neurones artificiels, d'injecter une base de données et de laisser tourner la << backprop >> pendant une nuit ; au matin, miracle, on retrouve une machine intelligente. Comme l'a dit Y. Le Cun, l'un des inventeurs de l'algorithme, l'usage de la RPG (rétropropagation du gradient) est à la fois *wide* et *wilde* (large et sauvage) ...

En fait, les performances de NetTalk étaient loin d'être exceptionnelles, si on les compare à ce qui se fait de mieux dans ce domaine de la lecture automatique. Il n'empêche que c'est une très jolie application, qu'on peut considérer comme le prototype de l'utilisation de la RPG pour un problème réel. Cette simulation démontre le pouvoir potentiel des réseaux de neurones : un temps de calcul raisonnable, une mise en oeuvre facile, et des performances acceptables. Mais elle montre aussi les limitations de l'approche : les performances ne sont *que* acceptables. >>

Les familles italo-américaines

Les deux arbres généalogiques ci-dessous présentent les relations entre les membres de deux familles comprenant chacune 12 personnes. On remarque que ces arbres sont isomorphes. Les relations sont : père, mère, mari, femme, fils, fille, oncle, tante, frère, soeur, neveu et nièce.

On souhaite faire apprendre ces relations à un réseau de neurones, c'est-à-dire que pour tout triplet de la forme (<personne1>, <relation>, <personne2>) décrit dans l'un des deux arbres, et toute entrée égale à (<personne1>, <relation>), le réseau calcule la réponse (<personne2>).

Figure 3.16 : Les familles américaines et italiennes de Hinton

Pour cela, Hinton utilise un réseau à 3 couches cachées dont l'architecture est décrite ci-dessous. Un groupe de 24 cellules d'entrée sert à coder les 24 personnes possibles. Un deuxième groupe de 12 cellules d'entrée sert à coder les relations. Chacun de ces groupes est connecté à un groupe de 6 cellules. Le rôle de cette couche est de coder l'information en entrée de manière optimale relativement au problème posé. La couche centrale contient 12 cellules ; c'est à ce niveau que la liaison personne-relation doit s'effectuer. L'avant dernière couche contient 6 cellules qui devra contenir une version codée de la sortie.

Figure 3.17 : Le réseau

Le réseau a été entraîné sur 100 des 104 relations possibles et après apprentissage prolongé, il a été capable de généraliser correctement sur les 4 exemples restants. Citons Hinton à ce propos :

<<It generalized correctly because during the training it learned to represent each of the people in terms of important features such as age, nationality, and the branch of the family tree that they belonged to, even these << semantic >> features were not at all explicit in the input or output vectors. Using these underlying features, much of the information about family relationships can be captured by a fairly small number of << micro-inferences >> between features. For example, the father of a middle-aged person is an old person, and the father of an Italian person is an Italian person. So the features of the output person can be derived from the features of the input person and of relationship. The learning procedure can only discover these features by searching for a set of features that make it easy to express the associations. Once these features have been discovered, the internal representation of each person (in the first hidden layer) is a distributed pattern of activity and similar people are represented by similar patterns. Thus the network constructs its own internal similarity metric. This is a significant advance over simulations in which good generalization is achieved because the experimenter chooses representations that already have an appropriate similarity metric>>.

Conclusion

Autant le perceptron est un dispositif très rudimentaire d'apprentissage, autant des algorithmes comme la rétropropagation du gradient appliqué à des réseaux multicouches permettent d'aborder des problèmes déjà très complexes. Parmi les applications les plus fréquentes de ces réseaux, on peut noter :

- **la reconnaissance des formes**. Il semble que ce soit là un des domaines où les réseaux neuronaux sont les plus performants. On peut signaler comme exemple un réseau reconnaissant les visages (voir ouvrage de Mitchell [Mit97]). c'est un exemple de solution connexionniste d'un problème pour lequel les méthodes classiques de l'intelligence artificielle ont été très peu performantes.
- **concurrence avec les méthodes statistiques**. Les réseaux neuronaux sont de plus en plus utilisés en marketing, scoring, ...avec des succès divers. D'après certains statisticiens, si ces nouvelles méthodes sont intéressantes et parfois plus performantes que les techniques statistiques usuelles, elles sont aussi moins robustes, moins bien fondées et partant, plus dangereuses.
- **la cognition**. L'espoir qu'ont suscité les techniques connexionnistes dans la communauté des sciences cognitives provient du fait que l'on a pensé avoir trouvé avec elles un dispositif expliquant ou montrant comment le << symbolique >> pouvait émerger spontanément de l'expérience. Le compte-rendu des familles de Hinton vont dans ce sens. Il me semble que les travaux et expérimentations visant à étudier ce phénomène n'avancent que très lentement.

Itération 2

QUOI : (définition, illustration)

Définition du connexionnisme :

- rendre compte des processus cognitifs à partir d'un ensemble d'unités, dotées chacune d'une faible puissance de calcul et interconnectées en réseau
- Discipline concernant les techniques de simulation des processus intelligents par des réseaux de neurones et des ordinateurs Neuronaux
- Le connexionnisme est un modèle de traitement de l'information

loi de Hebb
apprentissage

connexionnisme : démarche consistant à vouloir rendre compte de la cognition humaine par des réseaux de neurones.

définition et l'étude de réseaux de neurones formels qui sont des réseaux complexes d'unités de calcul élémentaire interconnectées

Un réseau de neurones formels est constitué d'un grand nombre de cellules de base interconnectées. De nombreuses variantes sont définies selon le choix de la cellule élémentaire, de l'architecture du réseau et de la dynamique du réseau.

Définition :

connexionnisme : démarche consistant à vouloir rendre compte de la cognition humaine (traitement de l'information) par des réseaux de neurones.

Un *réseau de neurones formels* est constitué d'un grand nombre de cellules de base, dotées chacune d'une faible puissance de calcul et interconnectées en réseau.

POURQUOI : (intérêt, histoire)

Première définition d'un neurone formel par McCulloch et Pitts en 1943
assemblée de neurones (Donald Hebb, 1949)
perceptron de Frank Rosenblatt (1958)
inventé globalement dans les années 1980 par Rumelhart et McClelland

applications des réseaux de neurones sont l'optimisation et l'apprentissage

- l'apprentissage supervisé ;
- l'apprentissage non supervisé ;
- l'apprentissage par renforcement.

Pour ces trois types d'apprentissage, il y a également un choix traditionnel entre :

- l'apprentissage << off-line >> : toutes les données sont dans une base d'exemples d'apprentissage qui sont traités simultanément ;

l'apprentissage << on-line >> : Les exemples sont présentés les uns après les autres au fur et à mesure de leur disponibilité.

Historique :

- Le neurone formel (McCulloch et Pitts en 1943)
- L'assemblée de neurones (Donald Hebb, 1949)
- Le modèle du Perceptron (Frank Rosenblatt, 1958)
- Perceptron multicouche (Rumelhart et McClelland, 1985)

Intérêt :

capacité d'apprentissage et d'optimisation :

- l'apprentissage supervisé ;
- l'apprentissage non supervisé ;
- l'apprentissage par renforcement.

un temps de calcul raisonnable, une mise en oeuvre facile, et des performances acceptables.

COMMENT : (fonctionnement, architecture, applications, limites)

s'opposent à l'approche classique des modèles symboliques

Le connexionnisme prend le contre-pied de l'approche symbolique en ce sens que les connaissances n'existent plus en tant que telles (plus de concepts, d'images mentales ou autre forme de connaissances), que les processus n'existent plus non plus et, troisième point, que les traitements seraient effectués en parallèle et non plus en séquentiel.

Modèles connexionnisme et symbolique ne sont pas opposés en réalité
interaction permanente entre hauts niveaux et bas niveaux

Une cellule élémentaire peut manipuler des valeurs binaires ou réelles. Les valeurs binaires sont représentées par 0 et 1 ou -1 et 1. Différentes fonctions peuvent être utilisées pour le calcul de la sortie. Le calcul de la sortie peut être déterministe ou probabiliste.

L'architecture du réseau peut être sans rétroaction, c'est à dire que la sortie d'une cellule ne peut influencer son entrée. Elle peut être avec rétroaction totale ou partielle.

La dynamique du réseau peut être synchrone : toutes les cellules calculent leurs sorties respectives simultanément. La dynamique peut être asynchrone. Dans ce dernier cas, on peut avoir une dynamique asynchrone séquentielle : les cellules calculent leurs sorties chacune à son tour en séquence ou avoir une dynamique asynchrone aléatoire.

Par exemple, si on considère des neurones à sortie stochastique -1 ou 1 calculée par une fonction à seuil basée sur la fonction sigmoïde, une interconnection complète et une dynamique synchrone, on obtient le modèle de Hopfield et la notion de mémoire associative.

Si on considère des neurones déterministes à sortie réelle calculée à l'aide de la fonction sigmoïde, une architecture sans rétroaction en couches successives avec une couche d'entrées et une couche de sorties, une dynamique asynchrone séquentielle, on obtient le modèle du Perceptron multi-couches (PMC) qui sera étudié dans les paragraphes suivants.

Il est bien évident que la plupart des problèmes d'apprentissage qui se posent naturellement ne peuvent pas être résolus par des méthodes aussi simples : il n'y a que très peu d'espoir que les exemples << naturels >> se répartissent << sagement >> de part et d'autre d'un hyperplan. Une manière de résoudre cette difficulté serait soit de mettre au point des séparateurs non-linéaires, soit (ce qui revient à peu près au même) de complexifier l'espace de représentation de manière à linéariser le problème initial. C'est ce que permettent de faire les réseaux multicouches que nous étudions maintenant.

En fait, les performances de NetTalk étaient loin d'être exceptionnelles, si on les compare à ce qui se fait de mieux dans ce domaine de la lecture automatique. Il n'empêche que c'est une très jolie application, qu'on peut considérer comme le prototype de l'utilisation de la RPG pour un problème réel. Cette simulation démontre le pouvoir potentiel des réseaux de neurones : **un temps de calcul raisonnable, une mise en oeuvre facile, et des performances acceptables**. Mais elle montre aussi les limitations de l'approche : les performances ne sont *que* acceptables.

The learning procedure can only discover these features by searching for a set of features that make it easy to express the associations. Once these features have been discovered, the *internal* representation of each person (in the first hidden layer) is a distributed pattern of activity and similar people are represented by similar patterns. Thus the network constructs its own internal similarity metric. This is a significant advance over simulations in which good generalization is achieved because the experimenter chooses representations that already have an appropriate similarity metric>>

la reconnaissance des formes, concurrence avec les méthodes statistiques, la cognition

Fonctionnement

- Pas de concepts, d'images mentales ou autre forme de connaissances : la représentation est distribuée sur les neurones (sous la forme d'un vecteur d'activation). Cette représentation est construite automatiquement par le réseau (sans influence de l'expérimentateur).
- Une cellule élémentaire (neurone) peut manipuler des valeurs binaires ou réelles. Le calcul de la sortie peut être déterministe ou probabiliste.
- Traitements seraient effectués en parallèle et non plus en séquentiel : toutes les cellules calculent leurs sorties respectives simultanément.

Architecture

- L'architecture du réseau peut être sans rétroaction, c'est à dire que la sortie d'une cellule ne peut influencer son entrée. Elle peut être avec rétroaction totale ou partielle.
- Modèle de Hopfield (mémoire associative) : neurones à sortie stochastique -1 ou 1 calculée par une fonction à seuil basée sur la fonction sigmoïde, une interconnexion complète et une dynamique synchrone.
- Perceptron multi-couches (PMC) : neurones déterministes à sortie réelle calculée à l'aide de la fonction sigmoïde, une architecture sans rétroaction en couches successives avec une couche d'entrées et une couche de sorties, une dynamique asynchrone séquentielle.

Applications

- reconnaissance des formes (exemple en lecture automatique avec NetTalk (rétropropagation de gradient RPG).
- traitement des données (type statistiques),
- la cognition

Limites

- Performances acceptables (mais non optimales)

Choix des illustrations

Définition :

connexionnisme : démarche consistant à vouloir rendre compte de la cognition humaine (traitement de l'information) par des réseaux de neurones.

Un *réseau de neurones formels* est constitué d'un grand nombre de cellules de base, dotées chacune d'une faible puissance de calcul et interconnectées en réseau. (Fig. 1 : Réseaux de neurones formels)

Historique :

- Le neurone formel (McCulloch et Pitts en 1943) (Fig 2a : Neurone formel)
- L'assemblée de neurones (Donald Hebb, 1949) (Fig. 2b : Loi de Hebb)
- Le modèle du Perceptron (Frank Rosenblatt, 1958) (Fig. 2c : Perceptron)
- Perceptron multicouche (Rumelhart et McClelland, 1985) (Fig. 2d : Perceptron multicouche)

Intérêt :

capacité d'apprentissage et d'optimisation :

- l'apprentissage supervisé ; (Fig. 3a. Apprentissage supervisé)
- l'apprentissage non supervisé ; (Fig. 3b. Apprentissage non supervisé)
- l'apprentissage par renforcement. (Fig. 3c. Apprentissage par renforcement)

un temps de calcul raisonnable, une mise en oeuvre facile, et des performances acceptables.

Fonctionnement

- Une cellule élémentaire (neurone) peut manipuler des valeurs binaires ou réelles. Le calcul de la sortie peut être déterministe ou probabiliste. (Fig. 5a. sortie binaire)
- Pas de concepts, d'images mentales ou autre forme de connaissances : la représentation est distribuée sur les neurones (sous la forme d'un vecteur d'activation). Cette représentation est construite automatiquement par le réseau (sans influence de l'expérimentateur). (Fig. 5b. Vecteur d'activation)
- Traitements seraient effectués en parallèle et non plus en séquentiel : toutes les cellules calculent leurs sorties respectives simultanément.

Architecture

- L'architecture du réseau peut être sans rétroaction, c'est à dire que la sortie d'une cellule ne peut influencer son entrée. Elle peut être avec rétroaction totale ou partielle.
- Modèle de Hopfield (mémoire associative) : neurones à sortie stochastique -1 ou 1 calculée par une fonction à seuil basée sur la fonction sigmoïde, une interconnexion complète et une dynamique synchrone. (Fig. 6a. Modèle de Hopfield)
- Perceptron multi-couches (PMC) : neurones déterministes à sortie réelle calculée à l'aide de la fonction sigmoïde, une architecture sans rétroaction en couches successives avec une couche d'entrées et une couche de sorties, une dynamique asynchrone séquentielle. (Fig. 6b. Perceptron multicouche cf. 2d)

Applications

- reconnaissance des formes (exemple en lecture automatique avec NetTalk (rétropropagation de gradient RPG). (Fig. 7. NetTalk)
- traitement des données (type statistiques),
- la cognition

Limites

- Performances acceptables (mais non optimales)

7 transparents à 2 mn par transparents = 15 mn

Trouver les figures, mettre au format, s'entraîner à présenter...