



HAL
open science

Architecture et neuro-mimétique

Bernard Allouche, Cyrille George, Pierre-Alain Jaffrenou, Hervé Lequay

► **To cite this version:**

Bernard Allouche, Cyrille George, Pierre-Alain Jaffrenou, Hervé Lequay. Architecture et neuro-mimétique. [Rapport de recherche] 0761/92, Applications et recherches en informatique et architecture (Laboratoire ARIA); Ecole Nationale Supérieure d'Architecture de Lyon; Bureau de la recherche architecturale (BRA). 1991. hal-03094576

HAL Id: hal-03094576

<https://hal.science/hal-03094576v1>

Submitted on 4 Jan 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CORDA 761

ÉCOLE D'ARCHITECTURE
DE
PARIS-BELLEVILLE

Ecole d'Architecture de LYON S

78. rue Rébeval
Tél. 42.41.33.60

Laboratoire ARIA

BIBLIOTHÈQUE

Applications et Recherches en Informatique et Architecture

Bernard Allouche - Cyrille George - Pierre Alain Jaffrenou - Hervé Lequay

Architecture et Neuro-mimétisme

Rapport de Recherche
Avril 1991

EA PARIS Belleville



0011632

Convention N° 90 01395 00 223 75 01

Ministère de l'Équipement, du Logement, des Transports et de la Mer
Direction de l'Architecture et de l'Urbanisme
Sous-Direction des Enseignements et des Professions
Bureau de la Recherche Architecturale

CORDA 761

(S)

N°Tnu 8464 - E

MFN 2567

CORDA

761

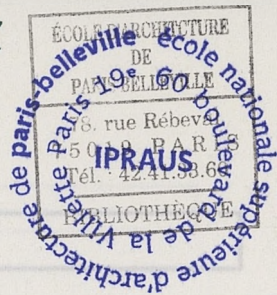


Table des matières

Introduction 1

« Le présent document constitue le rapport final d'une recherche remise au Bureau de la Recherche Architecturale, en exécution du Programme Général de Recherche mené par le Ministère de l'Équipement et du Logement avec le Ministère de la Recherche. Les jugements et opinions émis par les responsables de la recherche n'engagent que leurs auteurs. »

Les réseaux neuro-mimétiques :

- Intelligence artificielle..... 2
- Approche symbolique, approche connexionniste..... 3
- Mémoire associative et distribuée 4
- Traitement séquentiel, traitement parallèle..... 5

Historique succinct..... 4

Théories des réseaux de neurones..... 5

- Principes de base 5
- Topologies de réseaux et paradigmes d'apprentissage..... 9
- Intérêts du neuro-mimétisme 14
- Neuro-sciences et neuro-mimétisme..... 16

Implantation des réseaux neuro-mimétiques..... 19

- Applications et recherches..... 19
- Simulateurs..... 21

Application en Architecture : L'analyse d'image 24

- Problématique retenue..... 25
- Approches non connexionnistes existantes..... 26
- Approches connexionnistes possibles 26
- L'approche hybride retenue 30
- Orientations 34

Application en Architecture : Aide à la décision 35

- Contexte 35
- Problématique..... 38
- Méthodes algorithmiques..... 38
- Apports possibles de l'approche connexionniste 39
- Hypothèses..... 41

Perspectives 49

Bibliographie 50

Annexe : Analyse d'images..... 51

Table des matières

Introduction	i
Les réseaux neuro-mimétiques : Généralités	1
Intelligence artificielle.....	1
Approche symbolique, approche connexionniste.....	1
Mémoire associative et distribuée	2
Traitement séquentiel, traitement parallèle.....	3
Historique succinct.....	4
Théories des réseaux de neurones.....	5
Principes de base	5
Topologies de réseaux et paradigmes d'apprentissage.....	9
Intérêts du neuro-mimétisme	14
Neuro-sciences et neuro-mimétisme.....	16
Implantation des réseaux neuro-mimétiques.....	19
Applications et recherches.....	19
Simulateurs.....	21
Application en Architecture : L'analyse d'image	24
Problématique retenue.....	25
Approches non connexionnistes existantes.....	26
Approches connexionnistes possibles	28
L'approche hybride retenue	30
Orientations.....	34
Application en Architecture : Aide à la décision.....	35
Contexte	35
Problématique.....	38
Méthodes algorithmiques.....	38
Apports possibles de l'approche connexionniste	39
Hypothèses.....	41
Perspectives	49
Bibliographie	50
Annexe : Analyse d'images.....	51

Introduction

Les outils informatiques dédiés à la représentation et à la communication du projet d'architecture ont fait l'objet de progrès considérables ces cinq dernières années. A l'inverse, les outils d'analyse, de conception et de simulation restent désespérément peu capables d'auto-vérification de la cohérence interne des données qu'ils manipulent. Le plus souvent ils se comportent comme des boîtes noires totalement soumises au bon savoir faire de l'utilisateur.

Par exemple n'importe quel architecte, d'un simple coup d'oeil saura relever telle aberration géométrique dans un plan ou une perspective tandis qu'à l'inverse la quasi totalité des modeleurs autorise sans sourciller la génération d'objets exotiques. C'est un peu comme si un logiciel de traitement de texte ne gérait pas la largeur des pages et autorisait l'écriture en dehors de la feuille; ou à un niveau élémentaire si un éditeur de partition n'était pas capable de vérifier la cohérence rythmique de la mesure.

Il ne s'agit, dans ces exemples, que d'opérations de bas niveau. Mais on retrouvera les mêmes insuffisances, dramatiquement plus gênantes à des niveaux plus complexes comme par exemple au plan fonctionnel, constructif etc ..

Un autre défaut majeur des logiciels actuels en architecture est leur très faible performance au niveau d'une saisie intelligente des données nécessaires à leur fonctionnement. Ici aussi il est possible d'effectuer un parallèle avec ce qui se passe dans le cas d'application spécifique à d'autres secteurs d'activité: si l'on reprend le cas des logiciels de traitement de texte il est remarquable de noter qu'il est possible aujourd'hui d'entrer du texte directement par scanérisation avec un très bon taux de réussite.

Cette nouvelle fonctionnalité des traitements de texte modifie déjà les pratiques du secrétariat dans des secteurs d'activité (comme la recherche par exemple) ou la re-frappe de texte représente une part importante du travail.

Une des voies possibles à la conception de logiciels plus "intelligents" est de coupler les techniques algorithmiques classiques avec les techniques de l'intelligence artificielle. De cette manière on peut espérer élaborer des logiciels n'assurant pas seulement des fonctionnalités d'ordre arithmétique et géométrique par les voies algorithmiques classiques mais qui satisfassent aussi à un ensemble de logiques prédéterminées.

Il existe des équipes de recherche préoccupées par l'introduction des techniques classiques de l'intelligence artificielle c'est à dire du calcul symbolique de la connaissance dans la résolution informatique de problèmes relevant de la conception architecturale. Cette voie, vraisemblablement prometteuse à terme, devrait permettre de renforcer les facultés d'auto vérification de cohérence interne des éléments manipulés directement ou indirectement par les algorithmes lorsque qu'ils sont identifiables et aptes au calcul symbolique.

Toutefois certaines classes de problèmes ne semblent pas devoir être résolu de cette manière. C'est le cas à priori de problèmes relevant de la perception ou de la décision, où la connaissance manipulée n'est ou non facilement identifiable ou non facilement modélisable.

C'est cette voie que se propose d'explorer la recherche en cours. Il s'agit en situation de conception architecturale d'imaginer, d'élaborer et de tester des procédures informatiques à base de réseaux de neurones susceptibles de traitement intelligent de l'information.

Ce rapport de recherche présente les premiers éléments d'une réflexion allant dans ce sens. Après un rappel sur la théorie des réseaux neuro-mimétiques et l'état de la question dans les domaines de l'industrie et de la recherche, deux approches sont présentées.

L'une relève de la recherche fondamentale dans le cadre de la reconnaissance des formes en géométrie 2D. Bien qu'elle pose de sérieux problèmes de puissance informatique cette voie de recherche pourrait contribuer dans un certain terme à modifier les fonctionnalités et l'ergonomie des modeleurs 3D.

L'autre approche, plus appliquée, consiste en la recherche d'un réseaux d'aide à la conception d'enveloppes de bâtiment. Il s'agit là d'une approche tout à fait originale par rapport à l'approche "système expert" plus classique, qui nous permettra de tester la pertinence du concept réseau en situation de conception.

Les réseaux neuro-mimétiques : Généralités

Intelligence artificielle

L'ordinateur et l'informatique classique ont été conçus pour effectuer rapidement des tâches élémentaires de tri et de calcul. Un ordinateur fera ces opérations bien plus rapidement que n'importe quel prodige. Par contre, l'homme conserve une supériorité indubitable dans toutes les tâches dont il ne peut expliciter clairement les règles d'exécution séquentielle. Bien sûr, il aurait fallu, pour exécuter ces opérations à l'aide d'un ordinateur séquentiel et des modes de traitement classique, pouvoir expliciter et décomposer ces opérations complexes en sous-tâches élémentaires.

La recherche en intelligence artificielle tente d'explicitier et de formaliser les connaissances humaines et leur mécanisme d'inférence, ceci afin d'imiter le raisonnement et le comportement humain. Ses objectifs sont d'obtenir des résultats similaires à ceux qu'obtiendrait un homme. L'objectif des premières recherches en intelligence artificielle était de modéliser les connaissances d'experts, dans des domaines aussi variés que le diagnostic médical, la recherche pétrolière ou la prévision financière.

Approche symbolique, approche connexionniste

En intelligence artificielle classique, la modélisation des connaissances se fait sous forme de *symboles* et de règles de manipulation de ces symboles (*règles d'inférence*). La difficulté principale de cette approche réside dans le temps passé par les cognitivistes à démonter les mécanismes d'appréhension et de résolution des problèmes que les experts des domaines d'applications envisagés mettent en œuvre, souvent de façon non explicite. D'autre part, puisque à chaque concept correspond une structure de données, se pose le problème de l'organisation de ces données lorsqu'elles sont nombreuses et variées. Comment résoudre le problème de cohérence, d'ajout, de suppression de données sans bouleverser la base de connaissances, si ce n'est en se limitant aux données essentielles, et en abandonnant les cas limites ?

Si les méthodes d'intelligence artificielle classique sont particulièrement bien adaptées à la manipulation de concepts de haut niveau, elles rencontrent donc beaucoup de difficultés dans le traitement des données brutes. Le connexionnisme apporte des solutions pour la manipulation des données élémentaires et de bas niveau, que les réseaux arrivent à traiter sans pour autant en perdre les nuances. La reconnaissance de formes, par exemple, nécessite un traitement de ce type, avant tout traitement d'un niveau conceptuel plus élevé.

L'approche connexionniste va tenter de simuler le raisonnement non plus en partant du niveau symbolique de l'information, mais en essayant de modéliser le mode de fonctionnement du cerveau. En effet, les sciences neuro-biologiques ont montré que le cerveau humain ne fonctionnait pas sur le modèle qu'utilisaient les systèmes experts : les connaissances ne sont pas localisées dans le cerveau à des emplacements précis, mais plutôt distribuées dans celui-ci. De la même façon, on ne peut montrer un déroulement séquentiel d'opérations qui pourrait s'apparenter à des règles de déduction. L'approche connexionniste espère donc qu'en reproduisant les éléments physico-chimiques du cerveau, il en sortira quelque chose simulant mieux le raisonnement humain. Cette relation entre les recherches en intelligence artificielle et les Neuro-sciences se retrouve dans le vocabulaire commun employé : on parlera souvent de *réseaux neuro-mimétiques* pour désigner les objets utilisés pour la mémorisation des connaissances et le traitement de l'information, de *neurones*, d'*axones* et de *synapses* pour désigner les unités de traitement et leurs liaisons.

Pour illustrer la différence entre les deux approches, on peut prendre l'exemple du jeu d'échecs. Un système expert, par l'approche symbolique classique, choisira une stratégie en prévoyant coup par coup l'évolution du jeu, selon les règles qui lui auront été inculquées par un expert. La puissance du système sera sa capacité à prévoir à plus ou moins longue échéance. Un maître d'échec, au contraire du système expert, ne va pas prévoir l'évolution du jeu et la progression des pièces, mais se déterminer en fonction de son expérience devant une situation qu'il perçoit de façon globale et intuitive. Sa puissance proviendra de sa capacité à reconnaître, dans la situation présente, des situations similaires qu'il a déjà rencontrées.

Mémoire associative et distribuée

Dans l'approche connexionniste, on considèrera la mémoire comme *associative*. La mémoire associative est une mémoire à laquelle on a accès par le *contenu*, et non par l'*adresse*.

Dans une mémoire par adresse, il suffit de savoir *où* se trouve l'information pour la retrouver. Chaque information est accessible de manière univoque. Par contre, dans le cerveau, comme dans une mémoire associative, l'information ne peut être localisée à des emplacements déterminés, mais chaque donnée est *distribuée* sur l'ensemble de la structure qui constitue la mémoire. La notion d'adresse n'existe pas. La seule façon d'accéder à une information est de donner une partie de son contenu, ou une information similaire. La mémoire associative ne fonctionne que par associations d'idées.

On peut formaliser la mémoire sous la forme $y = Ax$, où A est une transformation faisant correspondre l'entrée x à une sortie y . Dans une mémoire par adresse, x est l'adresse et y la donnée placée en x . Dans une mémoire associative, x peut être un mot clé associé à y , ou une partie de y , ou encore une information à laquelle aura été associée y .

Quels peuvent être, à première vue, les avantages de la mémoire associative ? D'une part, la capacité à raisonner par analogie et à généraliser : à causes similaires, effets similaires. D'autre part, la capacité à raisonner par association, et donc, par extension, à reconstituer une information à partir de données partielles ou bruitées. Enfin, et non des moindres, l'avantage d'être robuste : la connaissance étant distribuée dans la mémoire, et non localisée, la destruction d'une partie de cette mémoire n'entraîne pas la perte de

L'information, comme c'est le cas pour une mémoire par adresse. Nous verrons à la fin du chapitre suivant l'étendue des intérêts de ce type de mémoire distribuée.

Historique succinct

Traitement séquentiel, traitement parallèle

La puissance du cerveau réside également dans sa capacité à traiter parallèlement les différents aspects de l'information. Par exemple, en reconnaissance de la parole, le cerveau intègre simultanément le signal sonore, la syntaxe, la sémantique, le contexte, des parties différentes du cerveau étant impliquées dans ces traitements. Comme nous le verrons ci-après, c'est le nombre de neurones impliqués dans le traitement qui importe, et non leur vitesse. Dans les ordinateurs séquentiels dont on dispose à l'heure actuelle, la vitesse de traitement est un palliatif au faible nombre de neurones que l'on peut simuler, en regard des quelques milliards que possède le cerveau humain. On peut attendre d'une simulation du comportement du cerveau qu'elle permette d'atteindre à coût faible cette puissance de traitement, exigée par les applications de reconnaissance de formes et de prise de décision.

En 1949, Hebb propose un modèle simple des mécanismes d'apprentissage, basé sur le renforcement ou l'affaiblissement des efficacités synaptiques.

A la fin des années 50, Widdrow et Hoff réalisent un adaptateur linéaire (*Adaline*) qui réalise une classification sur ses entrées. Mais sa capacité est limitée par le fait que les exemples présentés doivent être linéairement séparables.

En 1962, Rosenblatt, Minsky et Papert mettent au point une machine capable d'apprentissage : le *Perceptron*. C'est un réseau monocouche, composé d'une «rétine», réseau de capteurs, d'une couche de cellules d'association, et d'une cellule de décision, qui est un *Adaline*. Le perceptron est utilisé avec succès dans des expériences de reconnaissance visuelle. Mais en 1969, Minsky et Papert sonnent le glas des travaux sur le connexionnisme lors de la publication de leur théorème de limitation du perceptron, qui démontre que cette machine ne sera jamais capable de reconnaître deux classes non linéairement séparables (et par exemple incapable de résoudre un problème du type XOR).

Les études sur le connexionnisme ne seront reprises que dans les années 80, lors de recherches entreprises en mécanique statistique, et notamment sur les milieux désordonnés : la méthode du *recuit simulé* résout nombre de problèmes d'optimisation combinatoire. Hopfield, en 82, décrit un réseau complètement connecté capable de *mémoire associative*.

En 1985, les limitations du perceptron sont levées, en lui intégrant plusieurs couches de traitement, et surtout en proposant une méthode d'apprentissage satisfaisante : la *rétro-propagation du gradient*, qui permet un apprentissage réparti sur l'ensemble des couches du réseau (Le Cun et Fogelman). Cette approche est très séduisante dans la mesure où elle reproduit les phénomènes délocalisés d'apprentissage, de reconnaissance et de mémorisation que l'on retrouve dans le cerveau humain.

Historique succinct

En 1943, W. Mc Culloch et W. Pitts décrivent un neurone formel basé sur leurs travaux sur les neurones biologiques. Ils considèrent qu'un neurone effectue une somme pondérée des influx qui lui proviennent par l'intermédiaire des synapses. Si cette somme dépasse un certain seuil, ce neurone s'active et transmet aux autres neurones un influx égal à son niveau d'activation.

En 1948, Von Neumann introduit les *automates cellulaires*, en s'inspirant du fonctionnement du système nerveux. Son réseau, constitué d'un tableau bidimensionnel de cellules, est capable d'auto-reproduction.

En 1949, Hebb propose un modèle simple des mécanismes d'apprentissage, basé sur le renforcement ou l'affaiblissement des efficacités synaptiques.

A la fin des années 50, Widdrow et Hoff réalisent un adaptateur linéaire (*Adaline*) qui réalise une classification sur ses entrées. Mais sa capacité est limitée par le fait que les exemples présentés doivent être linéairement séparables.

En 1962, Rosenblatt, Minsky et Papert mettent au point une machine capable d'apprentissage : le *Perceptron*. C'est un réseau monocouche, composée d'une « rétine », réseau de capteurs, d'une couche de cellules d'association, et d'une cellule de décision, qui est un Adaline. Le perceptron est utilisé avec succès dans des expériences de reconnaissance visuelle. Mais en 1969, Minsky et Papert sonnent le glas des travaux sur le connexionnisme lors de la publication de leur théorème de limitation du perceptron, qui démontre que cette machine ne sera jamais capable de reconnaître deux classes non linéairement séparables (et par exemple incapable de résoudre un problème du type XOR).

Les études sur le connexionnisme ne seront reprises que dans les années 80, lors de recherches entreprises en mécanique statistique, et notamment sur les milieux désordonnés : la méthode du *recuit simulé* résout nombre de problèmes d'optimisation combinatoire. Hopfield, en 82, décrit un réseau complètement connecté capable de *mémoire associative*.

En 1985, les limitations du perceptron sont levées, en lui intégrant plusieurs couches de traitement, et surtout en proposant une méthode d'apprentissage satisfaisante : la *rétro-propagation du gradient*, qui permet un apprentissage réparti sur l'ensemble des couches du réseau (Le Cun et Fogelman). Cette approche est très séduisante dans la mesure où elle reproduit les phénomènes délocalisés d'apprentissage, de reconnaissance et de mémorisation que l'on retrouve dans le cerveau humain.

Dans les pages qui suivent, nous analyserons d'abord le modèle général du neurone et des connexions, puis les différentes topologies et méthodes d'apprentissage testées à ce jour.

Théories des réseaux de neurones

Pour définir le modèle général du réseau neuro-artificiel, nous utiliserons l'approche de Rumelhart et Hinton [RUM 89] et du groupe de recherche sur les processus distribués parallèles du MIT.

Le modèle connexionniste met en œuvre huit éléments principaux :

- un ensemble d'unités de traitement (les neurones eux-même)

Principes de base

L'architecture du réseau choisie dépendra de la fonction qu'on souhaite lui voir réaliser. Un réseau est défini par sa *topologie*, les caractéristiques des *neurones*, et les méthodes d'*apprentissage*. Les neurones peuvent être disposés de différentes façons : en grille, en étoile, en ligne, ou en plusieurs couches. Mais quelle que soit son architecture, l'élément constitutif de base, le neurone, est un simple automate. Il reçoit des signaux des autres automates auxquels il est relié, et produit, en fonction de son état antérieur et/ou de ses entrées, un signal de sortie qui deviendra son nouvel état. Ce signal sera à son tour envoyé vers les autres automates auxquels il est connecté.

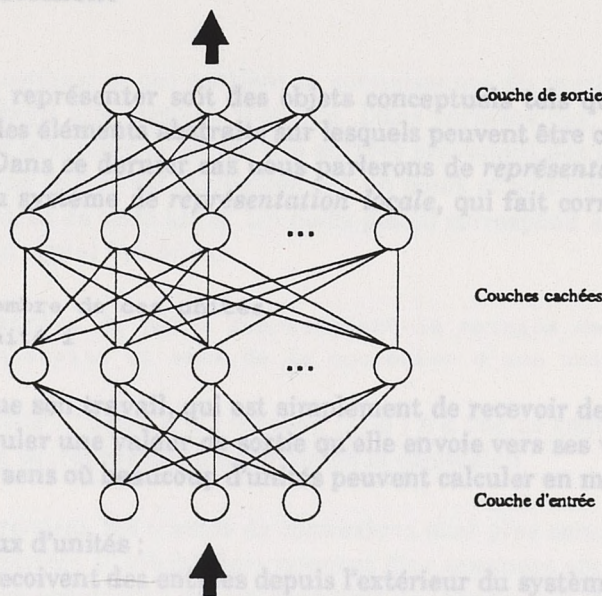
Les unités de traitement

Ces unités peuvent représenter des concepts, des attributs, lettres, mots, concepts ; ou des éléments auxquels peuvent être construites des structures significatives. Dans ce cas, nous parlerons de *représentation distribuée*, qui doit être opposée au *concept unique*, qui fait correspondre à chaque unité un concept.

Chaque unité effectue des calculs et reçoit des entrées depuis ses voisins, et d'en calcule les résultats qu'elle envoie vers ses voisins. Le système est parallèle dans le sens où toutes les unités peuvent calculer en même temps.

Trois types principaux d'unités :

- *input* : reçoivent des entrées depuis l'extérieur du système
- *output* : envoient des signaux à l'extérieur du système
- *hidden* : leurs entrées et sorties sont internes de l'intérieur du système.



Réseau multicouches

Les *connexions* (les synapses) ne sont pas figées. Chacune d'elle a un poids, fixe ou variable dans le temps, qui va pondérer le signal qu'elle transmet. De la sorte, certaines liaisons entre neurones sont renforcées, mettant en évidence des chemins privilégiés de l'information. Les méthodes d'apprentissage vont modifier les poids des connexions inter-neuronales de façon à permettre au réseau de réagir correctement aux entrées, et à produire le signal de sortie correspondant.

Dans les pages qui suivent, nous analyserons d'abord le modèle général du neurone et des connexions, puis les différentes topologies et méthodes d'apprentissage testées à ce jour.

Formalisation du réseau

Pour définir le modèle général du réseau neuro-mimétique, nous utiliserons l'approche de Rumelhart et Hinton [RUM 86] et du groupe de recherche sur les processus distribués parallèles du MIT.

Le modèle connexionniste met en œuvre huit éléments principaux :

- un ensemble d'*unités de traitement* (les neurones eux-même)
- un *état d'activité*
- une *fonction de sortie* pour chaque unité
- un *schéma de connexions* entre les unités
- une *règle de propagation* (des activités)
- une *règle d'activation* (combinaison des entrées d'une unité et de son état courant)
- une *règle d'apprentissage* (comment les connexions sont modifiées par l'expérience)
- un *environnement* dans lequel le système opère.

Les unités de traitement

Ces unités peuvent représenter soit des objets conceptuels tels que attributs, lettres, mots, concepts ; ou des éléments abstraits sur lesquels peuvent être construites des structures signifiantes. Dans ce dernier cas nous parlerons de *représentation distribuée*, qui doit être opposée au système de *représentation locale*, qui fait correspondre à chaque unité un concept.

N nombre de ces unités
 u_i unité i

Chaque unité effectue son travail, qui est simplement de recevoir des entrées depuis ses voisines, et d'en calculer une valeur de sortie qu'elle envoie vers ses voisines. Le système est parallèle dans le sens où beaucoup d'unités peuvent calculer en même temps.

Trois types principaux d'unités :

- *input* : reçoivent des entrées depuis l'extérieur du système
- *output* : envoient des signaux à l'extérieur du système
- *hidden* : leurs entrées et sorties proviennent de l'intérieur du système.

L'état d'activité

L'état du système à l'instant t est représenté par un vecteur de réels :

$a(t)$ instantané des activités de l'ensemble des unités
 $a_i(t)$ activité à l'instant t de l'unité i

L'état d'activité peut être continu ou discret. S'il est continu, il peut être ou non borné (par ex. de 0 à 10). S'il est discret, il peut être binaire (par ex. -1 ou 1) ou prendre ses valeurs dans un petit ensemble de valeurs.

La fonction de sortie

Les unités interagissent en transmettant des signaux à leurs voisines. L'intensité du signal est déterminée par leur degré d'activité.

$$o_i(t) = f_i(a_i(t))$$

$o_i(t)$ signal de sortie de l'unité i à l'instant t
 $f_i(a_i(t))$ fonction de sortie

Dans certains cas la fonction f peut être la fonction identité $f(x) = x$; dans d'autres, f est une *fonction à seuil*, une unité n'ayant aucun effet sur ses voisines si son activité n'excède pas un certain seuil ; dans d'autres, f est une *fonction stochastique* où le signal de sortie dépend de l'activité de l'unité de manière probabiliste.

Le schéma de connexion

C'est ce schéma de connexion qui contient la connaissance du système et qui détermine comment celui-ci réagit aux signaux d'entrée. Dans certains cas, l'entrée totale d'une unité du système est la somme pondérée des entrées provenant des unités auxquelles elle est connectée : chaque entrée est multipliée par un poids et additionnée aux autres pour former le signal d'entrée de cette unité. Un poids positif correspond à un signal *excitateur*, un poids négatif à un signal *inhibiteur*.

W matrice de poids représentant la matrice de connexion
 w_{ij} intensité et sens de la connexion d'une unité j vers une unité i

w_{ij} est positif si j excite i ; négatif si j inhibe i ; 0 si j n'a pas de connexion directe avec i . La valeur absolue de w_{ij} représente l'intensité de la connexion.

Dans le cas le plus général, les trames de connexions sont plus complexes : par exemple, les connexions excitatrices et inhibitrices peuvent être sommées séparément. Le schéma de connexion est alors représenté par un ensemble de matrices W , une pour chaque type de connexion.

La règle de propagation

La règle de propagation permet de calculer l'entrée *nette* d'une unité, à partir du vecteur de sortie $o(t)$ des autres unités.

$$net_i(t)$$

$net_e = W_e o(t)$ vecteur des entrées excitatrices
 $net_i = W_i o(t)$ vecteur des entrées inhibitrices

La règle d'activation

La règle d'activation définit comment les entrées nettes d'une unité sont combinées entre elles et avec l'état courant de l'unité pour produire un nouvel état d'activation.

$$a(t+1) = F(a(t), \text{net}(t))$$

$$a(t+1) = W_0(t) = \text{net}(t) \quad \text{si } F \text{ est la fonction identité et que l'état courant n'intervient pas}$$

F peut être une *fonction à seuil*, où l'entrée nette doit excéder un certain seuil pour modifier l'état d'activation. Parfois, l'état d'activation décroît lentement dans le temps, plutôt que de tomber à 0 quand il n'y a pas d'entrée externe. Dans d'autres cas, lorsque $a(t)$ peut prendre des valeurs continues, F peut être une fonction *sigmoïde* : chaque unité sature et atteint un minimum ou un maximum.

La classe de fonctions d'activation la plus commune est la fonction *quasi-linéaire* :

$$a_i(t+1) = F(\text{net}_i(t)) = F(\sum w_{ij} o_j)$$

La règle d'apprentissage

Cette règle modifie le schéma de connexion en fonction de l'expérience.

Trois types de modifications sont possibles :

- développement de *nouvelles connexions*
- *perte de connexions* existantes
- *modification des intensités* des connexions existantes

La création ou la perte de connexions peuvent être considérées comme des cas particuliers de la modification des intensités : si le poids d'une connexion devient non nul, elle est créée ; si ce poids devient nul, elle est détruite.

Virtuellement, toutes les règles d'apprentissage peuvent être considérées comme des variantes de la règle de Hebb [HEB 49], inspirée des travaux des neurophysiologistes :

Si une unité i reçoit un signal d'une unité j , et que les deux sont très actives, alors le poids w_{ij} de j vers i doit être renforcé.

$$\Delta w_{ij} = g(a_i(t), t_i(t)) h(o_j(t), w_{ij}) \quad t \text{ étant l'activation attendue}$$

Dans les versions les plus simples, il n'y a pas de superviseur dictant les réponses attendues du réseau, et les fonctions g et h sont proportionnelles à leur premier argument :

$$\Delta w_{ij} = \eta a_i o_j \quad \eta \text{ constante représentant le taux d'apprentissage}$$

Une autre variation commune est la règle de *Widrow-Hoff* [SUT 81], autrement appelée *règle delta*, car le niveau d'apprentissage est proportionnel à la différence entre la valeur d'activation actuelle et l'activation attendue donnée par le superviseur (target activation).

Dans ce cas :

$$\Delta w_{ij} = \eta (t_i(t) - a_i(t)) o_j(t)$$

C'est une généralisation de la règle d'apprentissage du perceptron.

Une autre règle (employée par Grossberg) :

$$\Delta w_{ij} = \eta a_i(t) (o_j(t) - w_{ij})$$

La représentation de l'environnement

Dans le modèle connexionniste, nous représenterons l'environnement comme une fonction stochastique variant dans le temps dans l'espace des motifs d'entrées possibles : à chaque instant, il y a une probabilité pour chacun de ces motifs possibles d'être l'entrée du système.

Généralement, l'environnement est caractérisé par une distribution stable des probabilités, indépendante des entrées passées et des réponses passées du système. Dans ce cas, nous pouvons représenter l'environnement comme un ensemble des entrées possibles $1..M$, caractérisées par une probabilité p . Chaque motif d'entrée étant un vecteur, nous décrirons cet ensemble comme *orthogonal* ou *linéairement indépendant*.

Certains modèles sont restreints aux types d'entrées qu'ils sont capables d'apprendre ; d'autres sont capables d'apprendre à répondre correctement seulement lorsque les vecteurs d'entrées forment un ensemble orthogonal ; d'autres uniquement si l'ensemble est linéairement indépendant ; d'autres encore sont capables d'apprendre à répondre à des motifs d'entrée arbitraires.

Topologies de réseaux et paradigmes d'apprentissage

Les réseaux sont déterminés par le type des automates qui les composent, mais aussi par leur capacité à apprendre. Certains réseaux n'apprennent pas, la topologie et le poids des connexions étant déterminés à l'avance et ne variant pas dans le temps. D'autres sont capables de modifier leur configuration par apprentissage. Jusqu'à présent, on ne conçoit que des réseaux dont les poids des liaisons synaptiques peuvent évoluer, la topologie et les fonctions de transition des automates restant constantes.

Les *réseaux figés* (dont les poids de connexion ne varient pas) sont utilisés comme outils d'aide à la décision ou d'optimisation. Ils reçoivent des stimuli de leur environnement, et après un temps transitoire, se stabilisent. Leur état est ensuite interprété comme une réponse du système à ces stimuli.

Les *réseaux évolutifs* sont doués d'apprentissage : la valeur du poids des connexions est modifiée lors d'une phase d'entraînement. Cet apprentissage peut être de deux types : l'*apprentissage supervisé*, où l'exemple et la réponse attendue sont donnés au réseau, afin de modifier les poids de connexion pour diminuer le taux d'erreur, et l'*apprentissage non supervisé*, où seul l'exemple est fourni au réseau. Dans ce cas, le système évoluant seul, on s'attend à ce qu'il classe lui-même les entrées, et qu'il soit apte à généraliser à partir des exemples fournis.

Apprentissage

La «connaissance» du réseau étant encodée dans les connexions entre cellules et dans les poids synaptiques, c'est du choix de ces connexions que dépendront les performances du réseau. Ces poids ne pouvant dans la plupart des cas être ni inventés, ni calculés, c'est la procédure d'apprentissage qui les créera.

Le principe de base d'un apprentissage est le suivant : on présente au réseau un ou plusieurs ensembles d'exemples, tout en activant la dynamique ou la méthode de relaxation du réseau ; on met en œuvre un algorithme de modification des poids permettant au réseau de s'adapter à l'environnement souhaité ; on boucle éventuellement sur la présentation des mêmes exemples, de façon à améliorer petit à petit les qualités de l'adaptation.

La modification du poids des connexions ne peut être que limitée à chaque essai, sous peine de voir le réseau perdre les informations déjà mémorisées. Cette modification est donc généralement fonction d'un *taux d'apprentissage*.

Apprentissage compétitif

Cette méthode d'apprentissage a pour but de mettre en compétition les unités regroupées en groupes, une seule unité d'un groupe pouvant être activée à un moment donné. L'apprentissage se fait en répercutant le poids des connexions non actives vers les connexions actives, le poids total étant limité à 1.

Une telle méthode permet de produire des détecteurs de figures ou des classifieurs très efficace. Si une structure est présente dans les stimuli d'entrée, les unités les sépareront en structures pertinentes. Autrement dit, le système trouvera des classes si elles existent. Plus les stimuli sont structurés, plus les classifications sont stables. Plus le nombre de groupes est grand, plus les classes détectées seront nombreuses.

Modèle de Hopfield

Le modèle de Hopfield met en œuvre un réseau totalement connecté (chaque unité est connectée à toutes les autres), et dont les unités reçoivent et produisent des valeurs binaires (0 ou 1, ou encore 1 ou -1). De plus, la matrice des poids est symétrique (le poids de connexion d'une unité i vers une unité j est le même que de j vers i).

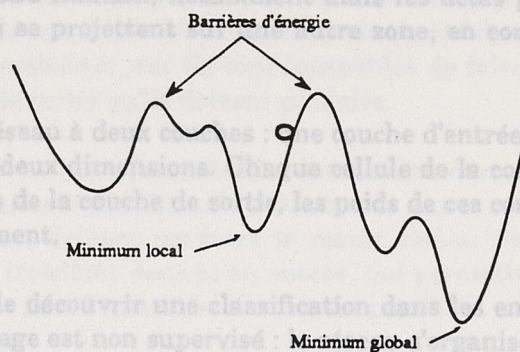
Le principe dynamique du réseau est une relaxation. La sortie de chaque unité i est envoyée vers toutes les autres unités du réseau, et calculée par une fonction à seuil. L'actualisation se fait en mode asynchrone (une unité tirée au hasard), et les propagations se poursuivent jusqu'à ce que les sorties ne soient plus modifiées, c'est à dire que le système converge vers une solution. Hopfield a démontré que cette convergence est garantie.

La phase d'apprentissage est statique, en mode supervisé : on présente un exemple en forçant les sorties, au temps 0, à correspondre au motif désiré. On applique une règle simple de Hebb, en augmentant le poids d'une connexion ij si les deux unités i et j sont

actives, en le diminuant dans le cas contraire.

La phase de généralisation (ou de reconnaissance) donne de bons résultats si les exemples appris sont des états stables de la dynamique du réseau. Une condition suffisante est que ces états soient orthogonaux deux à deux. On démontre que le nombre de prototypes mémorisés ne peut dépasser $0.14 \times N$, où N est le nombre d'unités employées.

On peut représenter la dynamique du réseau comme un «paysage d'énergie», surface souple dans laquelle les prototypes mémorisés formeraient des creux, des bassins d'attractions. L'entrée d'une nouvelle information revient à lâcher une bille sur cette surface. La dynamique du réseau tend à entraîner cette bille vers un creux de la surface. Par contre, le minimum atteint peut n'être que local, et non global, ce qui signifie que le réseau se stabilise à une solution correcte, mais non optimale. De plus, il peut exister des états stables parasites qui ne correspondent à aucun état appris.



Paysage d'énergie

Recuit simulé

Pour éviter de voir le système se stabiliser dans des états d'énergie qui ne représenteraient pas des solutions optimales, Kirkpatrick [KIR 83] proposa une méthode inspirée de la technique du recuit utilisée en cristallographie pour obtenir des matériaux très purs.

Elle revient à amener le système à un état d'agitation suffisant pour faire sortir la bille des minima locaux, puis à «refroidir» lentement le système. La probabilité pour la bille de tomber dans un minimum global est plus forte que celle de retomber dans un minimum local.

Machine de Boltzmann

La machine de Boltzmann a été introduite en 1984 par Hinton et Sejnowski [HIN 84]. Le modèle s'inspire à la fois des réseaux de Hopfield, du recuit simulé, et des réseaux multicouches, contenant à la fois des cellules visibles et des cellules cachées.

L'apprentissage dans les machines de Boltzmann met en œuvre une technique d'apprentissage-désapprentissage : dans une première phase, on présente au réseau des stimuli, ce qui aura pour effet de générer une surface d'énergie possédant des minima locaux marqués, et dans la seconde phase (dite de relaxation), le réseau fonctionne seul, en réduisant ces minima. Une règle peut permettre, dans cette phase de relaxation, de ne combler que les minima les moins profonds. Cette technique d'apprentissage reproduit les

processus observés dans le cortex, où le sommeil produit une agitation aléatoire, entraînant une sorte de désapprentissage.

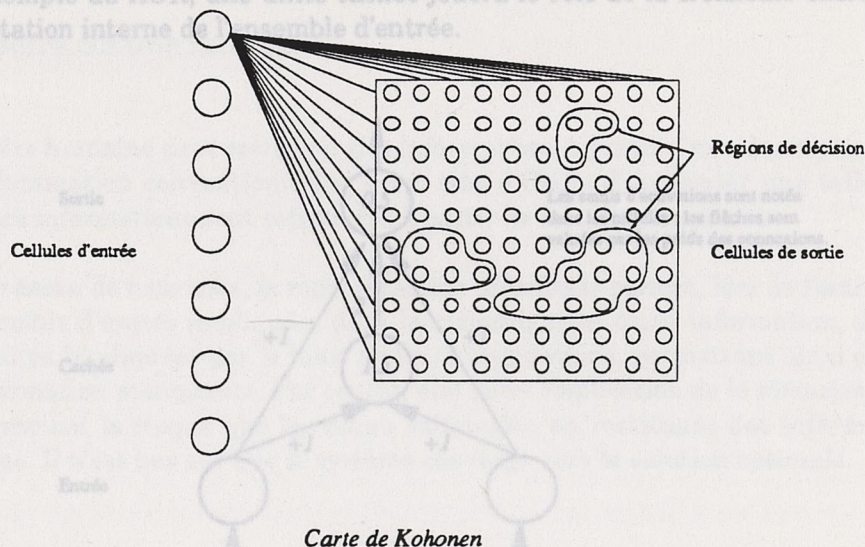
Intuitivement, on peut considérer les deux phases comme suit : pendant la phase contrainte, la performance du réseau est déterminée à la fois par les connexions existantes et par l'environnement. Pendant la phase libre, sa performance n'est plus déterminée que par les connexions internes, ce qui revient à soustraire des performances globales du système les performances dues aux interconnexions.

Cartes de Kohonen

Kohonen [KOH 84] proposa en 1984 un modèle de réseau inspiré des phénomènes mis en évidence dans le cerveau humain, notamment dans les actes perceptifs : les neurones d'une zone du cerveau se projettent sur une autre zone, en conservant les relations de voisinage.

Le modèle utilise un réseau à deux couches : une couche d'entrée, la rétine, et une couche de sortie en tableau à deux dimensions. Chaque cellule de la couche d'entrée est connectée à toutes les cellules de la couche de sortie, les poids de ces connexions étant variables, et initialisés aléatoirement.

Le réseau est chargé de découvrir une classification dans les entrées qui lui sont proposées, mais l'apprentissage est non supervisé : le réseau s'organise spontanément en *carte topologique*, et chaque région du réseau répondra à une classe des entrées. Toute la difficulté sera l'interprétation de ces cartes topologiques, pour trouver les classes que le réseau a découvert, et les zones correspondant à ces classes.



La dynamique utilisée consiste à renforcer les cellules sensibles aux entrées, tout en renforçant aussi celles qui lui sont proches. La distance entre l'unité la plus sensible et celles qui profitent de son activation décroît avec le temps. Le réseau est figé lorsque cette distance est nulle.

Le modèle a prouvé son intérêt dans des problèmes de reconnaissance de la parole (en classifiant en phonèmes), dans des problèmes d'appariement d'images, d'optimisation, etc.

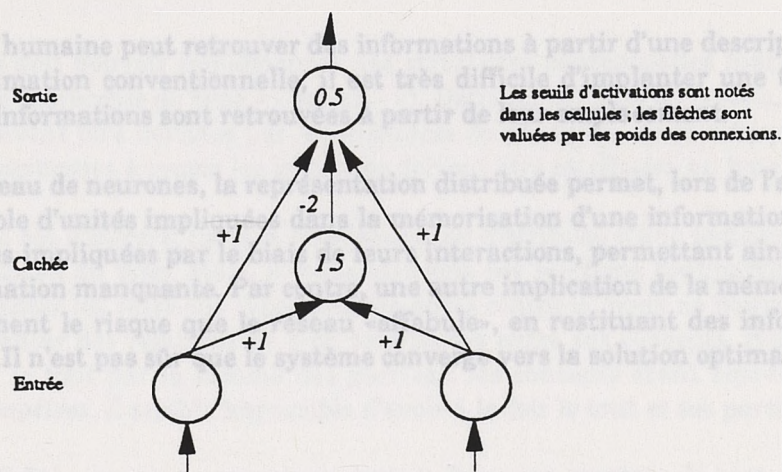
Perceptron multicouches et Rétro-propagation du gradient

Le perceptron tel que mis au point par Rosenblatt consistait en une rétine, une couche de cellules d'association, et une ou plusieurs cellules de décision. L'apprentissage, supervisé, modifiait les poids des connexions en fonction de la différence entre réponse attendue et réponse donnée, en fonction d'un facteur de gain.

Dans ces réseaux, il n'existe pas de *représentation interne* des stimuli. De tels réseaux sont capables de généralisation s'il existe une similarité de structure entre les figures d'entrée et les figures de sortie. Mais lorsque cette similarité de structure n'existe pas, ces réseaux ne peuvent fonctionner, car ils sont incapables de faire le lien entre les figures d'entrée et les figures de sortie qu'ils doivent produire.

Minsky et Papert [MIN 69] ont démontré les limitations des perceptrons monocouches, par exemple pour résoudre le problème du XOR, cas de figure où les deux figures d'entrée se ressemblant le moins doivent produire la même sortie. Une solution du problème consiste à fournir une troisième donnée en entrée, qui permettra d'introduire un discriminant supplémentaire dans l'ensemble des entrées possibles.

Par contre, si le réseau possède une couche cachée, connectée correctement aux unités d'entrée, on peut toujours trouver une représentation (un codage) interne qui permettra la correspondance entre entrée et sortie. Cette couche cachée modifie suffisamment les similarités des figures d'entrée pour permettre l'apprentissage des solutions. En reprenant l'exemple du XOR, une unité cachée jouera le rôle de la troisième entrée, et d'une représentation interne de l'ensemble d'entrée.



Une solution multicouches pour le problème du XOR

La règle d'apprentissage (type règle delta) des réseaux sans couches cachées est simple à mettre en œuvre - le changement de poids est simplement un pourcentage entre la valeur

attendue et la valeur produite. Par contre, dans le cas de réseaux avec couches cachées, il ne sera possible de propager l'erreur par une règle delta que si la fonction d'activation est semi-linéaire, c'est à dire une fonction non décroissante et différentiable de la sortie nette des unités auxquelles une unité est connectée. Une fonction à seuil linéaire ne peut convenir, sa dérivée étant infinie au seuil.

L'erreur détectée lors de l'apprentissage sera propagée récursivement des unités de sortie vers les unités d'entrée (d'où le terme de *rétro-propagation du gradient*), en calculant la dérivée de la fonction d'activation. Ainsi, de proche en proche, on modifiera les poids des connexions entre toutes les couches du réseau.

Si l'algorithme de rétro-propagation du gradient (encore appelé GBP - Gradient Back Propagation) a des qualités qui lui permettent de s'adapter à une très large classe de problèmes, il présente quelques défauts : la convergence peut être très longue ; les paramètres sont délicats à ajuster ; les résultats dépendent fortement de l'architecture du réseau et des valeurs initiales des poids de connexion. De plus, sa mise en œuvre semble souvent relever de l'expérience et de l'intuition.

Intérêts du neuro-mimétisme

Dans la section ci-après, nous reprendrons en les détaillant les principaux intérêts du modèle connexionniste et des mémoires associatives, tels que nous les avons évoqués au début de ce rapport.

La mémoire comme inférence

La mémoire humaine peut retrouver des informations à partir d'une description partielle. En programmation conventionnelle, il est très difficile d'implanter une telle mémoire, puisque les informations sont retrouvées à partir de leur emplacement.

Dans un réseau de neurones, la représentation distribuée permet, lors de l'activation d'un sous-ensemble d'unités impliquées dans la mémorisation d'une information, d'activer les autres unités impliquées par le biais de leurs interactions, permettant ainsi de reconstituer l'information manquante. Par contre, une autre implication de la mémoire distribuée est évidemment le risque que le réseau «affabule», en restituant des informations non demandées. Il n'est pas sûr que le système converge vers la solution optimale.

Similarité et Généralisation

Quand une nouvelle entité est mémorisée par un réseau, les modifications apportées aux connexions ne doivent pas détruire d'autres entités mémorisées. Ceci peut être évité en ne modifiant que très peu un grand nombre de poids, ou encore en utilisant un ensemble orthogonal de figures à reconnaître. Par contre, cette dernière technique fait perdre au

réseau un des avantages de la représentation distribuée, la *généralisation*, qui permet de reconnaître une situation même si celle-ci est légèrement différente.

Création de nouveaux concepts

Lorsque la représentation est locale (une unité / une entité), l'apprentissage de nouveaux concepts ne peut se faire, si le réseau est préconnecté, qu'en trouvant une unité libre, déjà connectée aux autres unités impliquées dans la mémorisation du concept.

En représentation distribuée, la seule chose à faire pour apprendre un nouveau concept est de modifier les connexions suffisamment pour créer une nouvelle figure d'activité stable. Le problème est alors, pour ne pas détruire des concepts déjà mémorisés, de trouver une figure d'activation qui modifierait au minimum les poids existant. Une solution est de faire "émerger" lentement le nouveau concept, plutôt que de le créer en une seule étape.

Résistance

Un autre avantage de la représentation distribuée est la résistance aux bruits et à la dégradation. Si les représentations des concepts sont non locales, une perturbation des connexions (par exemple, un bruitage) n'interdira pas la reconstitution de ces concepts. De la même façon, le réapprentissage après perturbation sera plus rapide, et le réseau sera capable de retrouver des entités pour lesquelles il n'a pas été réentraîné.

Représentation structurée

Un système doit pouvoir implémenter des structures conceptuelles hiérarchiques (par exemple une hiérarchie de type « - est un - »). Dans un tel type de réseau, les propriétés des *types* doivent être héritées par les *instances* de ces types, de la même façon que des propriétés communes à toutes les instances doivent être attribuées au type.

Une solution est d'inclure, dans la représentation distribuée d'une instance, la représentation distribuée du type. Mais, si nous utilisons ces relations tout/partie entre figures pour représenter les relations type/instance entre concepts, nous ne pouvons les utiliser pour représenter des relations de type tout/partie entre ces concepts. Nous ne pouvons représenter le tout par la somme des parties : les concepts étant représentés par des figures *alternatives*, il semble impossible d'avoir à la fois le tout et ses parties.

Hinton [HIN 81] a proposé une autre solution. Dans son système, le tout est composé de parties qui jouent un rôle particulier à l'intérieur de la structure du réseau. Chaque entité est représentée par une paire partie/rôle dans le tout; le type est représenté par un ensemble de ces paires. Dans ce cas, le tout n'est pas représenté par la somme des parties, puisque on introduit, en plus de la description des parties, la description du rôle que tiennent ces parties dans le tout.

Traitement séquentiel de symboles

Même avec la technique décrite précédemment, il apparaît qu'une seule structure hiérarchique peut être activée à la fois, si chaque groupe d'unités représente un rôle et que l'activité de ce groupe représente l'identité du constituant qui joue ce rôle. On retrouve cette contrainte dans le comportement humain. La représentation symbolique y est identifiée par les états successifs d'un vaste réseau interactif.

L'intérêt de la représentation distribuée, et de la représentation de structures hiérarchiques, est que le système peut retrouver l'identité d'un constituant à partir de la représentation du tout et de la représentation du rôle de ce constituant, et inversement, qu'une structure plus large peut être reconnue à partir de représentations de couples identité/rôle.

Les neurones communiquent

Neuro-sciences et neuro-mimétisme

Les neuro-sciences et l'intelligence artificielle sont étroitement connectées dans la mesure où les réseaux sont largement inspirés des études comportementales et physiologiques, et que les modèles de réseaux neuro-mimétiques ont permis la validation des hypothèses des biologistes sur le fonctionnement du cerveau. Le but du jeu étant de reproduire le raisonnement humain, puisqu'il est encore largement plus puissant que les outils imaginés à ce jour, il est nécessaire de garder en mémoire certaines des connaissances acquises en neurobiologie.

Les neurones sont lents

Au vu de la vitesse de propagation de l'information dans le cerveau, la plupart des tâches qu'il réalise (retrouver de l'information, généraliser, traiter du langage, etc) sont réalisées en moins de 100 pas. Des tâches similaires sur un ordinateur doivent donc être réalisées en moins de 100 itérations : c'est la contrainte du «programme de 100 pas». De plus, il ne semble pas qu'un neurone calcule des fonctions très complexes (activation, sortie). Dès lors, c'est le parallélisme massif qui donne au cerveau sa puissance.

Il y a un très grand nombre de neurones

De la même façon, chacun des 10^{10} neurones du cerveau est une unité de traitement active. Le parallélisme ne peut se comprendre que comme mettant en jeu un très grand nombre de ces unités de traitement.

Par contre, le nombre de neurones est limité. Donc, vu le nombre de tâches que doit réaliser le cerveau, chaque tâche ne peut mettre en jeu qu'un nombre limité de neurones. L'évaluation de modèles de réseaux doit prendre en compte cette contrainte.

Les neurones sont hautement interconnectés

Chaque neurone du cortex peut recevoir et transmettre des informations de 1 000 à 100 000 autres neurones par l'intermédiaire des synapses. L'activité d'un neurone dépend donc de l'activité d'un grand nombre d'autres neurones. Les décisions sont donc le fait de la *coopération* d'un grand nombre d'unités ; c'est une sorte de processus statistique.

L'apprentissage est la modification des connexions

La connaissance est dans les connexions plutôt que dans les unités de traitement. L'apprentissage se produit donc par une modification de la force des connexions.

Les neurones communiquent

Les neurones communiquent entre eux en s'envoyant des messages d'activation et d'inhibition. Les messages que peuvent s'envoyer les neurones ne peuvent être que des nombres signés de précision limitée, et en aucun cas des messages hautement symboliques. Les symboles, s'ils existent, ne peuvent émerger que de ce niveau *sub-symbolique* d'excitation et d'inhibition [HOF 79].

Les connexions du cerveau ont une structure géométrique et topologique

Les contraintes géométriques et topologiques sont grandes dans le cerveau, les connexions entre neurones étant relativement courtes. De plus, il existe une symétrie générale des connexions : entre deux régions du cerveau, s'il existe des connexions dans un sens, il en existe dans l'autre sens.

La rétro-propagation de l'erreur utilise ce retour. Dans d'autres modèles, on utilisera des connexions inhibitrices à l'intérieur d'une région, et excitatrices entre deux régions. La structure géométrique du réseau semble évidente, si l'on considère que les neurones ne peuvent être reliés, et donc envoyer des messages, qu'à des neurones relativement proches.

L'information est disponible continuellement

Les neurones sont constamment actifs, et envoient constamment des informations en sortie, même sans messages d'entrée. Si une unité représente une hypothèse, et son niveau d'activation le degré d'évidence de cette hypothèse, alors son activité donne des informations continues sur l'évaluation de cette hypothèse.

Dégradation proportionnelle au dommage

Implantation des réseaux neuro-mimétiques

Contrairement à la programmation conventionnelle, où la destruction d'une instruction ne permet plus le bon déroulement des processus de traitement, la destruction de connexions ou d'unités en représentation distribuée n'entraîne pas de dégradation brutale des performances, parce que le système est hautement redondant. Il n'y a pas de point critique dans la courbe de dégradation. La perte de performance du système est proportionnelle à l'importance du dommage.

Nous présenterons dans ce chapitre un panorama des applications existantes des réseaux de neurones formels. Le marché de ces applications est vaste, et touche des domaines très variés. Aussi ne listerons-nous que ceux qui paraissent les plus intéressants, en recherche envisagée, en reconnaissances de formes ou en aide à la décision.

Contrôle distribué

D'après nos connaissances du fonctionnement du cerveau, il n'y a pas de centre exécutif qui régirait le flux de traitement, sauf pour certaines tâches de bas niveau comme les fonctions vitales. Dans les tâches de plus haut niveau, plusieurs régions du cerveau jouent un rôle, s'influencent les unes les autres, et un dommage sur une de ces régions n'est pas crucial dans la performance globale du système.

La relaxation est le mode dominant de traitement

Le traitement d'informations semble être de chercher à résoudre un large ensemble de contraintes, et donc de «tomber» dans une solution plutôt que de «calculer» une solution. Le fonctionnement du système s'apparente plus à une réaction à un ensemble de données, à une relaxation suite à un influx.

Le domaine de la reconnaissance de forme est le secteur où le plus grand nombre d'applications ont été réalisées. On tire alors parti de la capacité des réseaux adaptatifs à éliminer le bruit de fond des signaux d'entrées, et de leur faculté à classifier ces entrées.

* Reconnaissance de caractères imprimés, de chiffres manuscrits :

Après une première phase de traitement d'image classique (détection de contours, masquage, filtrage), les signaux sont transmis à un réseau multicouche dont les cellules de décision s'activent proportionnellement au taux de reconnaissance du caractère. Une application développée par IBM permet la relecture de facture et autres documents depuis un scanner, en présentant une très faible sensibilité aux bruits (décalages, papier froissé, mauvaise typographie).

* Reconnaissance de visages :

Un modèle proposé, basé sur le modèle cognitif de mémoire des visages, identifie les visages en tenant compte du contexte. C'est un réseau multicouche, recevant ses entrées d'une rétine de 40x50 cellules, utilisant l'algorithme de rétro-propagation du gradient.

* Traitement d'image et Reconnaissance de formes :

Les réseaux sont utilisés pour améliorer la qualité des images en phase de pré-traitement (élimination du bruit), pour la compression / décompression, pour l'extraction de caractéristiques. En phase de reconnaissance, les réseaux servent à la recherche de frontières

Implantation des réseaux neuro-mimétiques

Un autre domaine privilégié d'application des réseaux neuro-mimétiques est l'aide à la décision. Le problème revient généralement à la résolution de contraintes multiples, de choix de priorités, de problèmes d'optimisation. Ces réseaux sont généralement adaptatifs, dans le sens où leurs connaissances sont réactualisées en fonction des situations rencontrées.

Nous présenterons dans ce chapitre un panorama des applications existantes des réseaux de neurones formels. Le marché de ces applications est vaste, et touche des domaines et des acteurs très variés. Aussi ne listerons-nous que ceux qui paraissent proche du domaine de recherche envisagé, en reconnaissances de formes ou en aide à la décision.

Une recherche sur les réseaux demande aussi de pouvoir implanter une architecture choisie, soit de façon logicielle, soit de façon matérielle. La deuxième partie de ce chapitre présentera nos travaux en matière de simulateurs de réseaux, ainsi qu'une liste non exhaustive des critères permettant le choix d'un simulateur du marché adapté aux besoins du laboratoire.

Applications et recherches

Le domaine de la reconnaissance de forme est le secteur où le plus grand nombre d'applications ont été réalisées. On tire alors parti de la capacité des réseaux adaptatifs à éliminer le bruit de fond des signaux d'entrées, et de leur faculté à classifier ces entrées.

- Reconnaissance de caractères imprimés, de chiffres manuscrits :

Après une première phase de traitement d'image classique (détection de contours, masquage, filtrage), les signaux sont transmis à un réseau multicouches dont les cellules de décision s'activent proportionnellement au taux de reconnaissance du caractère. Une application développée par IBM permet la relecture de facture et autres documents depuis un scanner, en présentant une très faible sensibilité aux bruits (décalages, papier froissé, mauvaise typographie).

- Reconnaissance de visages :

Un modèle proposé, basé sur le modèle cognitif de mémoire des visages, identifie les visages en tenant compte du contexte. C'est un réseau multicouches, recevant ses entrées d'une rétine de 40x50 cellules, utilisant l'algorithme de rétro-propagation du gradient.

- Traitement d'image et Reconnaissance de formes :

Les réseaux sont utilisés pour améliorer la qualité des images en phase de pré-traitement (élimination du bruit), pour la compression / décompression, pour l'extraction de caractéristiques. En phase de reconnaissance, les réseaux servent à la recherche de frontières

entre régions de textures distinctes, et donc à l'extraction de contours (en cartographie par exemple), à la détection de mouvements, à l'appariement d'images.

Un autre domaine privilégié d'application des réseaux neuro-mimétiques est l'aide à la décision. Le problème revient généralement à la résolution de contraintes multiples, de choix de priorités, de problèmes d'optimisation. Ces réseaux sont généralement adaptatifs, dans le sens où leurs connaissances sont réactualisées en fonction des situations rencontrées.

- Aide à la décision :

Des applications destinées aux milieux financiers permettent, en fonction de multiples critères, de décider de l'octroi de prêts, ou encore de prévoir des évolutions probables des marchés financiers. Il semble que de tels réseaux pourraient voir leurs domaines d'application se diversifier rapidement, dans tous les secteurs où les critères de décisions sont trop nombreux pour être appréhendés simultanément par un individu seul. Nous verrons dans le chapitre suivant comment ces réseaux pourraient être mis à contribution dans des logiciels de CAO.

- Organisation de locaux :

Une application élaborée par IBM permet l'organisation et l'implantation d'usines. Son but est de résoudre les problèmes de relation entre ateliers, en vue d'optimiser les temps de transferts. Le programme calcule les flux, regroupe les opérations fortement liées, implante les stations et postes de travail. La technique utilisée met en œuvre le recuit simulé.

Comme nous l'avons vu au début de ce rapport, le connexionnisme s'oppose à l'intelligence artificielle dans le sens où il tend à privilégier *l'intuition* et non le *raisonnement*. Si les systèmes experts de l'intelligence artificielle classique excellent à manipuler des symboles, ils rencontrent des difficultés dans le raisonnement associatif, alors que la démarche d'un expert se situe aux deux niveaux à la fois. D'où l'idée d'introduire des réseaux neuronaux dans les systèmes experts. Les applications privilégiées de ces systèmes experts connexionnistes sont celles qui traitent des problèmes de classification : recherche de diagnostics, détection de pannes, reconnaissance de formes.

Si ces systèmes n'ont pas encore montré un gain de performance significatif, ils sont cependant plus adaptatifs et plus rapides, ce qui les destine particulièrement à des applications en temps-réel.

Les recherches actuelles visent à améliorer la performance globale des réseaux neuro-mimétiques, aussi bien en terme de vitesse de traitement, qu'en terme d'accroissement de la fiabilité et des capacités de mémorisation.

On a vu que la capacité formidable du cerveau humain dans le traitement des informations résidait avant tout dans son parallélisme massif. Les architectures utilisées jusqu'à présent sont des architectures séquentielles. Nombre de recherches visent à implanter, de façon matérielle, une architecture parallèle, où des processeurs multiples travailleraient indépendamment les uns des autres (Connection Machine, Transputers, HyperCube, etc).

Mais l'intégration matérielle d'un grand nombre de neurones se heurte à des problèmes de place, d'échauffement des circuits, et de gestion des communications entre ces neurones. Aussi d'autres approches tendent à remplacer les circuits classiques (VLSI ou WSI) par des circuits optiques, en remplaçant les fils par des faisceaux lumineux, qui peuvent se croiser sans interagir. De plus, on pourra se servir d'hologrammes pour le stockage des informations (l'hologramme peut être considéré comme l'équivalent optique du réseau de neurones, l'information étant répartie dans l'ensemble du système).

Les recherches portent aussi sur l'amélioration ou la recherche d'algorithmes plus performants. Une recherche d'IBM étudie les neurones à fonction d'activation fractales, qui ont comme intérêt principal de permettre au réseau de visiter (et d'éviter) les minima locaux.

Simulateurs

L'intérêt premier d'un simulateur de réseaux de neurones est de pouvoir tester rapidement différentes topologies et architectures pour la résolution d'un problème donné. Un simulateur logiciel aura comme avantage sa facilité de description du réseau, sa souplesse d'utilisation et son confort de visualisation des résultats. L'étape suivante consistera à améliorer les performances du système, soit en fixant son architecture dans un composant logiciel optimisé, soit en le transférant en produit matériel. En effet, un programme séquentiel simulant un modèle connexionniste impose des temps d'apprentissage et de reconnaissance très longs.

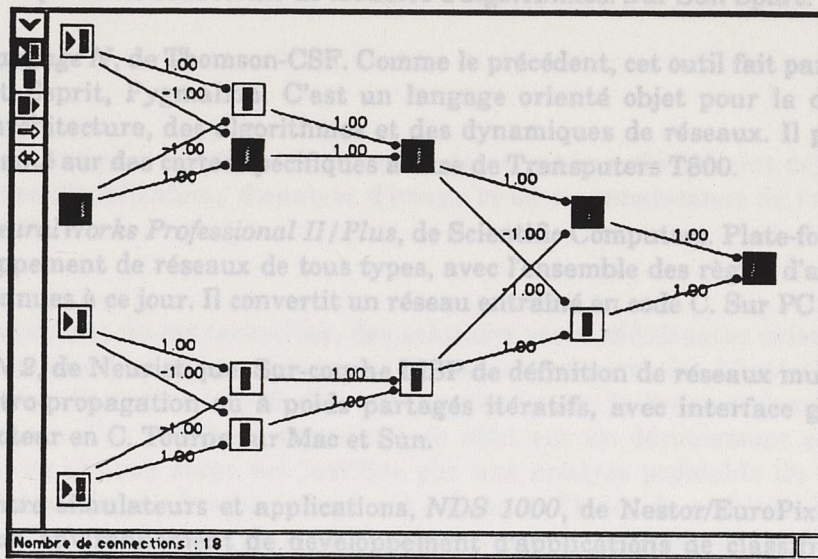
Pour un laboratoire de recherche, un simulateur doit être un outil extrêmement souple, capable de supporter a priori tous les modèles existants, mais aussi de permettre d'implanter des variations autour de ces modèles, et d'expérimenter des modèles nouveaux. Le meilleur simulateur se présenterait donc sous la forme d'un composant logiciel, dont les sources seraient accessibles, auquel on pourrait adjoindre des modules extérieurs, et posséderait tout l'enrobage destiné à en faciliter l'interfaçage avec le logiciel hôte.

Un simulateur devrait avoir les fonctionnalités suivantes :

- l'architecture du réseau peut être quelconque ;
- chaque unité du réseau peut recevoir un nombre arbitraire d'entrées, avec possibilité de regroupement et de traitement différencié de celles-ci ;
- les fonctions d'activation, d'entrée, de sortie, d'apprentissage peuvent différer d'un neurone à l'autre ;
- les unités peuvent être regroupées en couches, en équipes ;
- la configuration du réseau peut être modifiée à tout moment ;
- la dynamique peut être synchrone ou asynchrone ;

Au-delà des capacités du noyau d'implémentation du réseau, l'interface du simulateur doit être souple, autoriser des modes d'action variés, aussi bien pour la saisie des paramètres que pour la visualisation des résultats. Les états successifs du réseau doivent pouvoir être analysés facilement, par une interface graphique.

Dans le but d'améliorer notre compréhension du modèle connexionniste et des algorithmes qui le sous-tendent, nous avons commencé l'écriture d'un simulateur sur micro-ordinateur Apple. Ce simulateur, doté d'une interface graphique conviviale, se présente sous la forme d'une application standard Macintosh, et permet de créer à la souris des réseaux de divers types.



Interface du simulateur ARIA

Le logiciel permet de définir des couches de neurones, en distinguant unités d'entrée, cachées et de sortie. Les connexions sont gérées au niveau de ces unités, ou globalement par couches. Chaque connexion peut voir son poids fixé d'entrée ou modifié en cours de traitement. Les fonctions d'activation sont choisies neurone par neurone.

Le simulateur permet de visualiser à chaque instant l'état des neurones, en couleur, et d'étudier ainsi les propagations des activités dans le réseau.

Même si le logiciel est écrit en langage objet, et donc facilement extensible, sa relative lenteur et son manque d'outils d'assistance le limitent à un rôle didactique. Il n'est pas approprié pour la définition de réseaux à nombre élevé de neurones ou de connexions. L'ampleur de la tâche (pour en faire un simulateur de haut niveau) nous a contraint à en abandonner le développement, pour ne pas nous détourner de notre premier objet d'étude. Dans nos développements ultérieurs, nous écrirons plutôt, en fonction des besoins, des modules logiciels spécialisés.

Il existe actuellement sur le marché quelques simulateurs qui paraissent relativement performants. La liste ci-dessous présente les simulateurs exclusivement logiciels, ainsi que les plateformes de développement.

- *Rochester Connectionist Simulator*, de l'université de Rochester (USA). Ecrit en C, c'est un outil flexible qui autorise l'adjonction de modules externes. Il peut être intégré à un autre logiciel. Sur stations Sun.

- *Neuralis*, de Tau Ceti. Simulation de réseaux multi-couches. N'utilise pour l'apprentissage que la rétro-propagation du gradient. Ce logiciel peut générer en Pascal ou C le code permettant d'implanter le réseau dans un logiciel externe. Tourne sur PC.
- *Galatea*, de Mimetics. Cette librairie C permet la spécification de réseaux et de leur interface, ainsi que la mesure de leurs performances. Les différents modèles sont présents sous forme de modules d'algorithmes. Sur Sun Sparc.
- *Langage N*, de Thomson-CSF. Comme le précédent, cet outil fait partie d'un projet Esprit, Pygmalion. C'est un langage orienté objet pour la définition de l'architecture, des algorithmes et des dynamiques de réseaux. Il peut être implanté sur des cartes spécifiques à base de Transputers T800.
- *NeuralWorks Professional II/Plus*, de Scientific Computers. Plate-forme de développement de réseaux de tous types, avec l'ensemble des règles d'apprentissage connues à ce jour. Il convertit un réseau entraîné en code C. Sur PC et Sun.
- *SN 2*, de Neuristique. Sur-couche LISP de définition de réseaux multi-couches à rétro-propagation ou à poids partagés itératifs, avec interface graphique et moteur en C. Tourne sur Mac et Sun.
- Entre simulateurs et applications, *NDS 1000*, de Nestor/EuroPixels. Il s'agit d'un environnement de développement d'applications de classification et de reconnaissance, basé sur l'utilisation dynamique d'unités constituées de trois couches, avec allocation dynamique des neurones et des connexions. A partir d'un jeu de données caractérisées, NDS génère le réseau, apprend, vérifie et diagnostique, et intègre le réseau et son run-time dans une application (avec apprentissage en ligne). Ce logiciel a permis le développement d'applications de tous types, en particulier de détection radar pour l'armée américaine.
- Dans le domaine pédagogique, le Groupe Neurones de l'Ecole Polytechnique Fédérale de Lausanne a développé un jeu de simulateurs spécialisés, qui permet de comprendre le fonctionnement de réseaux de type Hopfield, Kohonen, rétro-propagation, Héroult-Jutten, pendule inversé. L'utilisateur visualise l'état des réseaux, peut interagir en modifiant les paramètres définissant les réseaux. Sur Sun Sparc.

La reconnaissance de visages étant un problème très complexe, on peut s'interroger sur les possibilités offertes par le connexionnisme en matière de reconnaissance de formes plus simples d'un point de vue géométrique, avec l'intention d'appliquer ces techniques au champ de l'architecture.

Application en Architecture : L'analyse d'image

Nous avons mentionné le fait que les réseaux de neurones artificiels ont déjà été étudiés dans le cadre d'applications d'analyse d'image et de reconnaissance de forme avec un certain succès.

Parmi les résultats de ces tentatives, des solutions assez séduisantes existent pour des problèmes très complexes (reconnaissance de visages par exemple). L'intérêt de ces solutions s'apprécie d'autant mieux si elles sont comparées aux techniques classiques de traitement d'image. Ces dernières s'appuient en effet sur un déroulement séquentiel du traitement, où chaque étape est justifiée par une analyse préalable du problème à résoudre. Cette analyse cherche notamment à ramener l'étude du problème initial à celle d'un ensemble de problèmes plus simples pour lesquels des solutions élémentaires de traitement d'image existent. Cette analyse peut s'appuyer par exemple sur des caractéristiques géométriques plus ou moins connues et constantes de l'image à traiter. La plupart des applications industrielles exige des temps de traitements très courts (contrôle de fabrication, pilotage temps réel de robots, comptage, etc...) et l'analyse doit permettre d'exploiter au mieux les caractéristiques géométriques de l'image et les techniques de traitement les mieux appropriées.

Lorsque l'approche précédente est possible, le traitement d'image offre des outils très puissants et de très bons résultats. Malheureusement, il existe des familles de problèmes pour lesquelles ces techniques sont inopérantes, ou tout au moins très difficiles à mettre en œuvre. La reconnaissance de visage est typiquement un problème où l'image ne possède pas de caractéristiques géométriques évidentes, tant les différences entre deux prises de vue d'un même visage peuvent être importantes. De plus la géométrie d'un visage est difficile à formaliser, rendant un découpage séquentiel du traitement pratiquement impossible. Dans ce cas, des approches statistiques existent mais sont peu fiables (grande sensibilité aux dégradations de l'image à traiter par exemple). L'approche connexionniste s'affranchit, nous l'avons vu, très bien des difficultés de formulation d'un problème sous forme séquentielle (programmation d'un algorithme) et donne, dans le cas précis de la reconnaissance de visage, de très bons résultats. De nombreuses expériences ont été faites par des équipes de recherche à travers le monde sur ce sujet et l'intérêt pour cette technique est grandissant (en France, l'équipe du laboratoire de recherche en informatique de Paris Sud (Orsay) travaille actuellement sur ce thème).

La reconnaissance de visages étant un problème très complexe, on peut s'interroger sur les possibilités offertes par le connexionnisme en matière de reconnaissance de formes plus simples d'un point de vue géométrique, avec l'intention d'appliquer ces techniques au champ de l'architecture.

Plusieurs applications peuvent en effet être envisagées, si l'on admet bien sûr leur faisabilité à plus ou moins long terme. Parmi celles-ci, on peut citer par exemple la classification de constructions au vu de leur caractéristiques architectoniques (époque, influences, etc...) à partir de documents photographiques. Le lien avec la reconnaissance de visage est ici assez fort pour que l'on puisse espérer pouvoir s'appuyer sur des démarches déjà éprouvées. Toutefois, il s'agit là d'un problème très difficile à traiter, de par la nature des informations fournies, et il est prématuré d'envisager un développement sur ce thème.

Il est aujourd'hui admis que la vision humaine fait participer environ 60 % du cortex et, à elle seule, la rétine de l'œil composée de 100 000 000 de récepteurs «élémentaires», auxquels s'ajoutent plusieurs couches de neurones, représente une puissance de calcul équivalente à plusieurs milliards d'opérations par seconde. Autant dire que la vision artificielle est un problème extrêmement complexe. Dans la pratique, il ne s'agit évidemment pas de reproduire le comportement d'un être humain face à une œuvre d'art, mais plutôt celui d'une mouche en plein vol. Des études ont montré une organisation et une complexité déjà étonnante pour le système visuel de la mouche. Son œil abrite en effet un réseau composé d'environ un million de neurones, commandant en temps réel les mouvements des ailes, en fonction des réponses de neurones détecteurs de mouvements (Deuxièmes Entretiens de Lyon, N. Franceschini, CNRS LNB3 Neurocybernétique, Marseille).

Aussi, nous sommes nous tournés vers une autre problématique liée à l'architecture, à savoir l'interprétation de documents graphiques (plans, esquisses,...) et qui s'apparente à la reconnaissance de formes simples d'un point de vue géométrique.

Ce type de problèmes a déjà fait l'objet de nombreux développements dans les domaines liés à l'intelligence artificielle symbolique, notamment la lecture automatique de plans par exemple, et plus particulièrement en architecture. L'objectif recherché est de proposer un outil de saisie automatique de documents 2D en amont d'un système de CAO. L'approche connexionniste nous intéresse de ce point de vue.

Problématique retenue

Dans la suite, nous nous proposons d'étudier la faisabilité d'un outil construit autour d'un réseau connexionniste et destiné à la reconnaissance de formes simples. Par formes simples, il faut entendre des formes planaires, dans une première phase, volumiques en projection perspective dans un deuxième temps, dont la définition géométrique est élémentaire: un parallélépipède ou un cube en perspective par exemple, dessinés à main levée ou saisis sur une tablette graphique.

Cette étude s'appuiera sur la présentation des méthodes existantes en traitement d'image, puis par la suggestion de méthodes connexionnistes qui paraissent intéressantes, et enfin en explicitant la démarche que nous aurons retenue.

Approches non connexionnistes existantes

... à familles a des avantages et
systématique et est assez rapide. En revanche, certaines configurations de contours
provoquent des erreurs :

Le spécialiste d'analyse d'image s'intéresse généralement aux images (informations en deux dimensions) dans lesquelles il recherche des objets simples pour les compter, déterminer leurs positions etc... Malgré cette première restriction sur l'information à traiter, la tâche reste assez complexe et l'on peut distinguer trois étapes de difficulté croissante :

- l'identification de formes planes disjointes ;
- l'analyse d'un ensemble de formes en vrac ;
- l'identification de volumes dans une image plane.

Le problème de l'identification de formes planes

d'erreur peut être évité, mais
il est aussi plus complexe et plus long : il tient davantage de la recette de cuisine que
d'une méthode systématique.

Les images considérées sont numériques, en principe bien contrastées et ne possèdent que des objets disjoints. Ces caractéristiques vont permettre de segmenter l'image en régions regroupant des pixels plus ou moins homogènes (par rapport à leur brillance par exemple). De nombreuses techniques de segmentation existent (binarisation, multi-seuillage, croissance de région, nuées dynamiques, maximisation de la variance inter-classe, ...). Une fois cette segmentation obtenue, on cherche à extraire des caractéristiques discriminantes pour chacune des régions, et déterminer ainsi s'il s'agit ou non d'un objet recherché. Dans des cas simples, ces caractéristiques peuvent être exprimées par des constantes (rapport périmètre/surface/diamètre pour un disque, moments d'inertie, etc...).

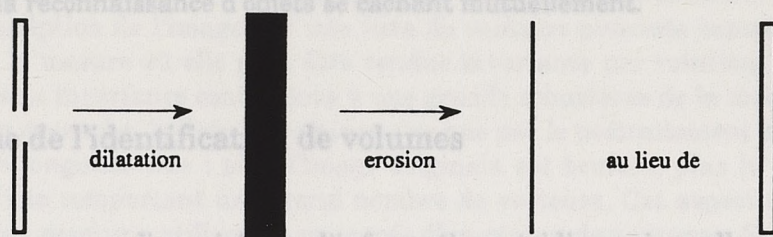
Parfois, ces critères ne sont pas suffisants ou n'existent pas. On doit alors passer à une description plus détaillée de l'objet à reconnaître. Cette description repose le plus souvent sur le contour de l'objet représenté par une ligne polygonale fermée. Se pose alors le problème de l'extraction des contours des objets et de la vectorisation de ces contours.

reconnaitre. Cette phase d'appariement est en fait un ensemble de règles de décisions,
pouvant être éventuellement intégrées à un système expert pour des images complexes

Plusieurs méthodes existent pour ce qui concerne l'extraction des contours (gradient, laplacien, opérateur de contraste, filtres de Sobel, Prewitt, etc...) mais ne fournissent que des descriptions «pixels» et non vectorielles des contours. Les opérateurs utilisés ont une grande sensibilité au bruit, au contraste de l'objet par rapport au fond, et la description «pixels» obtenue est en niveaux de gris. Cette information sur le contour doit donc être binarisée (pour lever l'ambiguïté des niveaux de gris, on transforme en tout ou rien : un pixel est ou n'est pas un pixel de contour, en fonction de son niveau de gris) .

A l'issue de ces traitements successifs, il n'est pas certain que le contour obtenu soit parfaitement continu : des perturbations de l'image, un mauvais contraste ou simplement la phase de binarisation peuvent entraîner des discontinuités plus ou moins nombreuses et importantes. Il s'agit alors de rendre le contour connexe. Là encore, plusieurs méthodes existent, que l'on peut regrouper en deux familles : celles faisant appel à des opérateurs morpho-mathématiques (érosion, dilatation), et celles recherchant dans un voisinage donné le pixel contour le plus proche (la notion de proximité évoque ici la distance sur la trame des pixels mais aussi un certain nombre de critères sur le contour lui-même: alignement des pixels, directions plus ou moins parallèles des alignements, etc...). De très nombreuses publications existent sur ce sujet: [MKE 85] fournit un exemple d'application

confrontée à ce problème de reconnexion. Chacune de ces deux familles a des avantages et des inconvénients bien sûr. La famille «morphologique» possède un caractère plus global, systématique et est assez rapide. En revanche, certaines configurations de contours provoquent des erreurs :



Pour la deuxième famille, le traitement est local, et ce type d'erreur peut être évité, mais il est aussi plus complexe et plus long : il tient davantage de la recette de cuisine que d'une méthode systématique.

Un ensemble de défauts communs à ces deux familles de méthodes peut être énoncé :

- comment choisir le seuil de distance au delà duquel deux segments ne doivent pas être reconnectés ;
- grande sensibilité au bruit ;
- inapplicables à des images grises.

Vient ensuite la phase de vectorisation. Celle-ci nécessite un prétraitement qui amincit le contour jusqu'à l'obtention d'une épaisseur d'un pixel. La vectorisation peut alors être menée dans les meilleures conditions, par un suivi pas à pas des pixels de contour. Pour tenir compte de la complexité de l'objet, cette vectorisation est associée à la construction d'un graphe qui permettra une interprétation symbolique de l'ensemble des vecteurs (on y retrouvera la notion d'angle, de sommets, d'arêtes, de points de concours de plusieurs vecteurs, etc...). Ce graphe servira au contrôle de l'existence de caractéristiques géométriques simples, et à la comparaison avec un modèle de référence représentant l'objet à reconnaître. Cette phase d'appariement est en fait un ensemble de règles de décisions, pouvant être éventuellement intégrées à un système expert pour des images complexes (robotique).

Pour ce qui concerne notre problématique, rien ne s'oppose a priori à l'élaboration d'un algorithme «classique» s'appuyant sur la démarche qui vient d'être détaillée. Il faudra cependant que le résultat fournit soit indépendant de l'orientation et du rapport d'homotétie qu'il peut exister avec le forme de référence. C'est sur cet aspect notamment que nous escomptons de bons résultats avec la méthode connexionniste.

Le problème d'un ensemble de formes en vrac

Dans l'énoncé du problème précédent, nous avons supposé que l'image présentait n objets disjoints les uns des autres. Les résultats sont alors obtenus en un temps proportionnel à n . Dans le cas contraire, les solutions à envisager sont plus complexes à mettre en œuvre et leur temps de réponse est alors exponentiel en n . Ces solutions s'appuient sur des descriptions des objets que l'on recherche dans l'image (à l'aide d'arbres du type précédent) et conjointement sur le principe d'une construction d'hypothèses et de leur vérifications à

posteriori. De nombreuses variantes existent et peuvent faire appel à des systèmes experts symboliques pour vérifier la cohérence de la forme.

Là encore, au vu des résultats sur la reconnaissance des visages, les réseaux neuro-mimétiques pourraient s'avérer plus efficaces que les méthodes classiques pour traiter le problème de la reconnaissance d'objets se cachant mutuellement.

Le problème de l'identification de volumes

Ce problème suppose l'acquisition d'informations tridimensionnelles. La saisie de l'information s'effectue à l'aide de capteurs spéciaux (ultra-son, laser, palpeurs,...) ou traditionnels (caméra stéréoscopique). La reconstruction d'images se faisant dans ce dernier cas par appariement d'images. Nous sortons ici de notre cadre de travail car nous voulons travailler sur des documents plans non stéréoscopiques (documents de type projection perspective).

Approches connexionnistes possibles

Nous venons de voir que les outils «traditionnels» d'analyse d'images offrent un ensemble de solutions répondant à notre problématique. Ces solutions s'articulent, de façon plus ou moins complexe, pour former un outil adapté au problème. Cet outil peut toutefois s'avérer très sensible à la qualité des données présentées en entrée (bruitage de l'image originale, déformation des objets à reconnaître, etc...), et même inadapté si l'énoncé du problème varie quelque peu. Ainsi, en d'autres termes, la robustesse de ces méthodes n'est généralement pas très bonne et il est très utile d'observer le comportement de solutions «neuro-mimétiques» de ce point de vue.

Pour cela, nous allons proposer différentes approches neuro-mimétiques en essayant de montrer leurs intérêts et en soulignant les difficultés qu'elles amènent. Ces approches diffèrent essentiellement par la nature de l'information qui leur est présentée. En effet, la mise en place d'un réseau connexionniste passe par une importante phase de réflexion portant sur le codage de l'information présentée en entrée du réseau. Ce codage peut faciliter la résolution du problème, et a contrario, un problème mal posé (un mauvais codage de l'information) peut entraîner l'échec d'une approche connexionniste.

Soit donc un problème à résoudre entrant dans le cadre de notre problématique: par exemple une image d'une définition de 500x500 pixels en niveaux de gris représentant un carré (tracé à main levée) fortement dégradé et numérisé à l'aide d'une caméra (mauvaises conditions de numérisation : éclairage, mise au point, etc...). Il existe deux façons de représenter l'information exprimant l'existence d'un carré dans l'image. La première sous forme de pixels : le carré est un ensemble de pixels plus ou moins homogènes. La seconde, est de considérer le carré sous la forme d'une liste de vecteurs obtenue à l'aide d'un prétraitement de l'image. Pour ces deux codages de l'information, nous allons suggérer des solutions neuro-mimétiques pour interpréter cette information codée et reconnaître la présence d'un carré.

L'approche par une liste de vecteurs

Nous savons comment, à partir d'une image en niveaux de gris, il est possible de représenter un contour à l'aide d'une liste de vecteurs, et quels sont les problèmes rencontrés lors de cette phase de vectorisation (notamment le problème de la discontinuité du contour). La description de l'image par une liste de vecteurs présente cependant un grand intérêt dans la mesure où elle peut être rendue invariante par rotation, translation ou homotétie. Cette invariance contribuera à une grande robustesse de la méthode de reconnaissance. En revanche, la liste de vecteurs obtenue par le prétraitement n'est pas nécessairement de longueur fixe : plus l'image originale est bruitée, plus le prétraitement fournit une liste comportant un grand nombre de vecteurs. Cet aspect est en fait une forte limitation pour une utilisation en entrée d'un réseau de neurones. En effet, pour un réseau de type multi-couches par exemple, il existe toujours une couche réservée à l'entrée des données, et celle-ci possède un nombre fixe de neurones. D'une façon générale, toute entrée sur un réseau se fait par l'intermédiaire de n neurones recevant chacun une partie de l'information : la valeur de n ne peut être supérieure à une valeur maximale n_0 , fixée dès la conception du réseau et fortement dépendante de l'architecture de celui-ci.

Le codage de l'information par une liste de vecteurs, s'il paraît intéressant d'un point de vue géométrique, semble donc impraticable par le fait qu'elle suppose une entrée de taille variable.

L'approche «bit map»

Pour traiter une information de taille fixe, il suffit de traiter l'image entière sous forme de pixels. Cette solution est celle adoptée pour la reconnaissance de visages. En effet, les informations élémentaires fournies au réseau sont dans ce cas les valeurs des niveaux de gris de tous les pixels de l'image. Mais dans cette application particulière, les images utilisées sont de petites tailles, de l'ordre de 32×32 ou 64×64 . Le réseau qui traite cette information se comporte alors comme une mémoire associative (adressable par une partie de son contenu). Pour de tels réseaux, il faut autant de neurones que de pixels, avec pour chacun d'entre eux des connexions vers tous les autres. Pour une image de 32×32 pixels, cela représente donc 1024 pixels, donc 1024 pixels et donc plus d'un million de connexions. Chaque connexion étant représentée par un poids codé sur un certain nombre de bits (par exemple huit), cela fait un quantité d'information énorme pour un dispositif capable de mémoriser des petites images. Par expérience, on sait que le nombre d'images pouvant être mémorisées est pratiquement proportionnel au nombre de connexions. Le réseau ainsi construit pourra ainsi stocker une centaine d'images (le nombre de pixels stockés est de l'ordre de 10% du nombre de connexions).

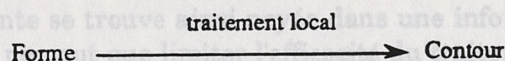
Considérons un tel réseau, entraîné à reconnaître soit un carré, soit un triangle, pourvu qu'ils soient tous deux inscrits dans une image de 32×32 pixels. Si l'on fait subir une rotation de 180 degrés au triangle ou une translation, le réseau ne sera pas capable de conclure efficacement, sauf s'il a également été entraîné sur ces variantes de présentation. Il faudra donc entraîner le réseau pour toutes les variantes de présentation, et le nombre d'images ainsi apprises va devenir important, saturant ainsi la capacité de mémorisation du réseau.

Dans notre cas, les problèmes de précision géométrique nous obligent à travailler sur des images d'une bonne définition (par exemple 500x500). Un réseau comparable au précédent du point de vue des fonctionnalités occuperait au minimum une taille mémoire de l'ordre de $(500 \times 500) \times (500 \times 500)$ octets, soit 62 500 000 000 octets. Un tel réseau est évidemment irréalisable. Nous abandonnerons donc définitivement cette approche globale.

L'approche par voisinage

Pour pallier cet inconvénient, on peut imaginer que l'on traite l'image par morceaux, en déplaçant une fenêtre, et en traitant le contenu de cette fenêtre à l'aide d'un réseau de neurones. C'est ainsi que J. Loncelle démontre dans [LON 89] la faisabilité de l'extraction de contours sur une image en niveaux de gris à l'aide d'un «pattern associator». Cette solution est intéressante dans la mesure où elle donne de bons résultats par rapport aux techniques classiques, mais avec l'inconvénient de temps de traitement très longs, défaut inhérent au handicap technologique dont souffre toute implémentation de réseau neuro-mimétique simulé.

Pour ce qui nous concerne, cette approche par balayage de voisinages, si elle limite la taille mémoire, risque de ne pas être adaptée à notre problème. En effet, un traitement de l'image par voisinage signifie que l'information possède un caractère local : le contour d'une forme est défini localement et ne dépend pas de l'aspect de l'ensemble contour de la forme. Nous avons donc la relation suivante :



Réciproquement, la connaissance locale d'un contour ne permet pas, sans faire d'autres hypothèses, de déterminer à quel type de forme le contour correspond. Il faut donc connaître l'ensemble du contour pour avoir une chance de reconnaître la forme correspondante. Cette contrainte peut alors devenir incompatible avec une technique de balayage, pour des contours non contenus dans le voisinage utilisé notamment.

L'approche hybride retenue

Il apparaît à la suite de ce qui vient d'être dit, que l'objectif de la reconnaissance de formes géométriquement simples sur une image à l'aide d'un réseau neuro-mimétique ne pourra être atteint sans que l'information à traiter soit de taille fixe et sous une forme suffisamment condensée. Des problèmes un peu similaires au notre ont déjà été résolus, notamment la reconnaissance de signaux sonar. Que ce soit à l'aide de sa transformée de Fourier ou de sa décomposition en ondelettes, le signal initial de longueur temporelle très grande est ramenée à un signal où l'information utile est plus «compacte». C'est ainsi que dans [THO 89] par exemple, un réseau identifie des signaux sonar brefs mêlés à un bruit de fond complexe.

Par analogie avec [THO 89], nous devons ramener notre problème non local (nécessité de traiter toute l'image) à un problème local à l'aide d'une transformation appropriée. Cette transformation revient à un codage de l'information initiale sous une forme pouvant être acceptée facilement par un réseau de neurones. On retrouve donc la phase importante et commune à toute approche neuro-mimétique, la recherche d'un bon codage de l'information à traiter.

Dans la suite, afin de valider notre codage, nous nous limiterons dans un premier temps à un sous problème plus simple : la recherche de segments de droite dans l'image. Puis nous en déduirons une méthode pour notre problématique initiale, la recherche de figures géométriques simples.

La recherche de segments de droites

Nous nous proposons donc de traiter une image numérique, saisie à partir d'une caméra ou d'un scanner en niveaux de gris, et représentant un document graphique comportant des segments de droites. Lors de cette phase d'acquisition, l'information numérique propre au document graphique va être mêlée à une autre information parasite qui va perturber la qualité de l'image numérique finale. Cette perturbation a plusieurs causes :

- niveau d'éclairage du document, contraste, etc...
- problème de l'éclairage uniforme du document à numériser
- défauts propres au dispositif de conversion numérique : bruit de fond, défauts de fabrication, etc...

L'information intéressante se trouve ainsi noyée dans une information parasite, qui, si elle n'est pas éliminée, ne peut que limiter l'efficacité du traitement par réseau de neurones. D'une façon générale, le neuro-mimétisme n'est pas une solution à tous les problèmes et doit être associé à des techniques classiques : plus nombreuses seront ces techniques associées dans un dispositif neuronal, plus grande sera la part de connaissance implicite, et plus facile sera la tâche d'élaboration du réseau.

Aussi nous nous efforcerons de garder le maximum de prétraitement sous forme algorithmique. Soit donc à traiter l'image 1 (voir annexe). Cette image a été obtenue à l'aide d'une caméra noir et blanc, à partir d'un tracé à main levée sur un papier blanc. On peut noter l'incidence des conditions d'éclairage sur le contraste de l'image et sur la dérive d'éclairage (image 2).

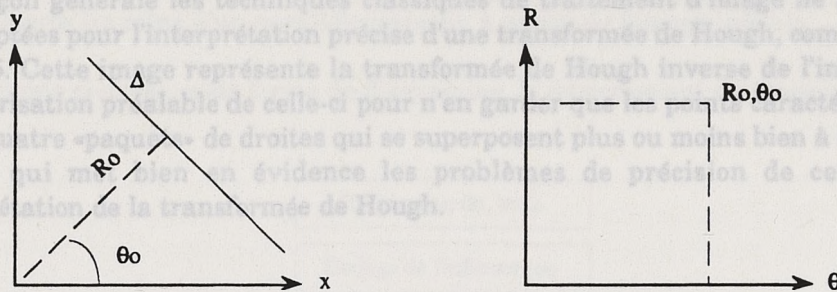
Cette dérive d'éclairage empêche notamment une binarisation de l'image pour l'extraction des «segments de droites». On doit donc utiliser un seuillage adaptatif pour cette tâche d'extraction. Un tel seuillage nous est fourni par les outils de la morphologie mathématique en niveaux de gris [SER 86], il s'agit de la transformation "Top Hat". Le résultat de cette transformation est ensuite binarisé pour donner le résultat de l'image 3. A ce stade, l'information est en bonne partie «nettoyée» du bruit ; il nous reste cependant à la condenser, pour l'appliquer en entrée d'un réseau.

Le codage de l'information que nous recherchons a pour but la transformation du problème global en problème local : l'image de départ contient une information étendue sur un voisinage d'une taille comparable à celle de l'image (le segment de droite), l'image transformée code cette information sur un voisinage très petit. Or il s'agit ici de détecter des segments de droite. La géométrie nous fournit un outil tout indiqué pour un tel codage ; il s'agit de la transformée de Hough.

Principe de la transformée de Hough

A une droite cette transformation fait correspondre un point, selon le principe suivant :

D'une façon générale les techniques classiques de traitement d'image ne sont pas très bien adaptées pour l'interprétation précise d'une transformée de Hough, comme le montre l'image 5. Cette image représente la transformée de Hough inverse de l'image 4, après une binarisation préalable de celle-ci pour n'en garder que les caractéristiques. On obtient quatre «paquets» de droites qui se superposent plus ou moins bien à l'image originale, ce qui confirme en évidence les problèmes de précision de cette méthode d'interprétation de la transformée de Hough.



Transformée de Hough de la droite Δ

Le point correspondant à la transformée de Δ a pour coordonnées R_0, θ_0 avec la relation suivante :

$$R = y \cos \theta + x \sin \theta$$

Il est intéressant de noter une propriété intéressante, et dont nous reparlerons un peu plus loin : si l'on considère l'ensemble des droites passant par le point de coordonnées (x, y) , la transformée de cet ensemble est une sinusoïde. Par conséquent, pour deux points P_1 et P_2 , on obtient deux sinusoïdes qui se coupent forcément en un point. Celui-ci correspond à la transformée de la droite passant par P_1 et P_2 . Pouvons nous pousser plus loin cette remarque en considérant non plus deux points, mais un segment de droite. Ce segment génère une famille de sinusoïdes concourant toutes en un point qui est la transformée de Hough de la droite support du segment. Ceci nous conduit à expliciter le principe de l'algorithme de la transformée de Hough, appliqué à la détection de segments de droites dans une image.

Cet algorithme repose tout d'abord sur une discrétisation en R et θ du problème avec un pas respectif ΔR et $\Delta \theta$, et sur l'utilisation d'une image binaire (un pixel est soit noir, soit blanc). Il utilise également une matrice $M[R, \theta]$, R variant de 0 à R_{\max} avec un pas de ΔR et θ variant de 0 à 180 degrés avec un pas de $\Delta \theta$. Cette matrice va être utilisée comme un accumulateur pour compter le nombre de fois où la droite R, θ est susceptible d'apparaître dans l'image à traiter. Voici l'algorithme simplifié correspondant :

```

pour tous les points P de l'image faire
  début
    pour q variant de 0 à 180 degrés avec un pas de Δq faire
      début
        calculer le R correspondant à la droite
        passant par P et de normale orientée q;
        incrémenter le compteur M[R, q];
      fin
    fin
  fin

```

Si l'on applique cet algorithme à l'image 3 on obtient le résultat illustré par l'image 4. Le pas de discrétisation est ici de un degré pour les angles et de un pixel pour les distances. Sur cette image on remarque un «paquet» de sinusoïdes, les points les plus sombres de

cette image sont regroupés en taches qui correspondent aux valeurs les plus fortes de l'accumulateur, et donc aux droites susceptibles d'être présentes dans l'image 3. La nature de ce résultat rend impossible la sélection d'un seul point dans une tache représentant une droite, par des techniques classiques (voir à ce sujet [TAG 89]).

D'une façon générale les techniques classiques de traitement d'image ne sont pas très bien adaptées pour l'interprétation précise d'une transformée de Hough, comme le montre l'image 5. Cette image représente la transformée de Hough inverse de l'image 4, après une binarisation préalable de celle-ci pour n'en garder que les points caractéristiques. On obtient quatre «paquets» de droites qui se superposent plus ou moins bien à l'image originale, ce qui met bien en évidence les problèmes de précision de cette méthode d'interprétation de la transformée de Hough.

C'est donc à ce stade, que nous faisons intervenir les techniques neuro-mimétiques. A l'heure où nous rédigeons ce rapport, l'architecture du réseau que nous utiliserons n'est pas encore définitivement arrêtée. Nous pouvons toutefois nous appuyer sur les travaux de J. Loncelle [LON 89] pour en donner une description rapide. Il s'agira en effet d'un réseau de type «pattern associator» en entrée duquel seront appliqués des voisinages de très petites dimensions qui balayeront la transformée de Hough. Le réseau réagira en fonction de la présence ou de l'absence dans le voisinage d'un point caractérisant une droite de l'image originale, de la même façon que le réseau [LON 89] réagit à la présence ou à l'absence de pixels contour. Ce réseau devrait donc, dans une tache de la transformée de Hough, être capable d'extraire un seul point représentant la droite la plus caractéristique de cette tache.

Notre méthode de reconnaissance de segments de droite s'appuie donc sur deux phases complémentaires qui en font une approche hybride :

- une phase de prétraitement et de codage du signal de l'image ;
- une phase d'interprétation par réseau neuro-mimétique.

Afin d'améliorer les temps de traitement, d'une part, de diminuer le diamètre des taches et d'en améliorer le contraste, d'autre part, on peut augmenter le pas de discrétisation. L'image 6 représente la transformée de Hough de l'image 3, mais avec un pas de cinq degrés pour les angles, et de 5 pixels pour les distances. L'image résultante possède donc 36 colonnes de 103 lignes contre 180 colonnes de 512 lignes pour l'image 4. On pourra, dans ce cas, se passer d'un balayage de voisinage, et traiter l'image de la transformée de Hough dans son ensemble.

En revanche, une analyse sémantique de l'image, nous paraît aujourd'hui trop prématurée : les réseaux de neurones étant des solutions de bas niveau et peu disposées à la

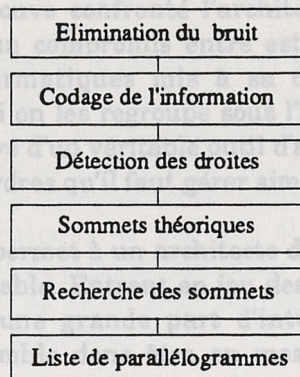
La recherche de polygones simples (parallélogrammes)

Dans l'approche précédente, nous nous sommes limités à la détection de droites supports de segments. Nous allons voir comment modifier cette méthode pour détecter des parallépipèdes.

Remarquons tout d'abord que, si les angles sont portés en abscisse et les distances en ordonnée, deux droites parallèles se transforment en deux points portés par la même verticale. De même, un parallélogramme se transformera en deux couples de points alignés sur deux verticales.

On pourra alors en ajoutant une couche précédente au réseau précédent, détecter sur la transformée de Hough la présence de parallélogrammes. En fait, ce réseau ne détectera que la présence de quatre droites support d'un parallélogramme. Il faudra alors vérifier l'existence effective de celui-ci. Pour cela, on recherchera les intersections des quatre droites dans l'image en examinant les voisinages des quatre sommets théoriques. Cette examen est un problème local, et peut être traité à son tour par un second réseau neuronal qui détecte les coins d'un parallélogramme, à partir de l'image initiale.

Nous donnons ci-après, sous forme de diagramme, le déroulement du traitement:



Le dispositif que nous venons de proposer est donc a priori capable de détecter les parallélépipèdes d'une image qui lui est présentée en entrée. Toutefois, la mise en œuvre des réseaux neuro-mimétiques reste à faire pour valider effectivement cette approche.

Orientations

Dans les phases suivantes nous finaliserons l'approche neuro-mimétique étudiée au chapitre "Architecture et neuro-mimétisme — L'analyse d'image". En outre, nous envisageons, en modifiant le dispositif, la détection de parallélépipèdes projetés en perspective cavalière. On peut en effet envisager la recherche, dans la transformée de Hough d'une image, de la trace de la transformée d'un tel volume représenté par ses arêtes.

En revanche, une analyse sémantique de l'image, nous paraît aujourd'hui trop prématurée : les réseaux de neurones étant des solutions de bas niveau et peu disposées à la manipulation de symboles de haut niveau.

Laisant de côté tous les critères subjectifs, nous admettrons qu'une enveloppe est globalement performante si :

- elle réalise une séparation efficace entre les milieux qui l'environnent,
- elle résiste aux contraintes que produisent ces milieux, de façon à ce que sa séparation reste suffisante tout au long de sa durée de vie.

A ces deux critères principaux, nous ajouterons celui du coût de revient de l'enveloppe, ainsi que le coût de maintenance éventuel.

La terminologie que nous emploierons dans la suite de votre étude est la suivante :

Application en Architecture : Aide à la décision

Les problèmes auxquels se trouve confronté l'architecte sont souvent des problèmes d'ordre décisionnel : réussir un compromis entre esthétique, performance, coût, etc. Actuellement, les outils informatiques mis à sa disposition se cantonnent dans l'assistance au dessin, même si on les regroupe sous l'appellation "conception assistée". La difficulté de la mise en œuvre d'un véritable outil d'aide à la conception réside dans la masse d'informations de tous ordres qu'il faut gérer simultanément.

D'autre part, le processus qui permet à un architecte de choisir entre des solutions multiples est difficilement modélisable. Entrent en jeu des critères éminemment subjectifs, impossibles à quantifier, et une grande part d'intuition et d'expérience cumulée. L'approche connexionniste semble donc être en mesure d'apporter des éléments de réponse à ces besoins d'outils décisionnels.

Notre but était de trouver un domaine restreint d'étude, pour mesurer la capacité des réseaux de neurones à traiter des problèmes rencontrés en conception architecturale. Ces réseaux, si leurs performances étaient prouvés, devraient pouvoir s'intégrer sans trop de difficultés dans la chaîne de production, au niveau de la conception elle-même. Suite à une collaboration avec les services d'expertise du Département d'Architecture de l'Ecole Polytechnique de Lausanne, où s'élaborent des outils logiciels d'assistance à la maintenance des bâtiments, nous avons orienté notre choix vers l'assistance à la conception de parois de bâtiments.

Contexte

Lorsqu'un architecte conçoit un bâtiment, il se trouve toujours confronté au choix des composants des enveloppes de ce bâtiment. Les critères qu'il doit prendre en compte pour opérer ce choix sont multiples : esthétique, durabilité de l'enveloppe, coût, rapidité et simplicité de mise en œuvre, performance, etc.

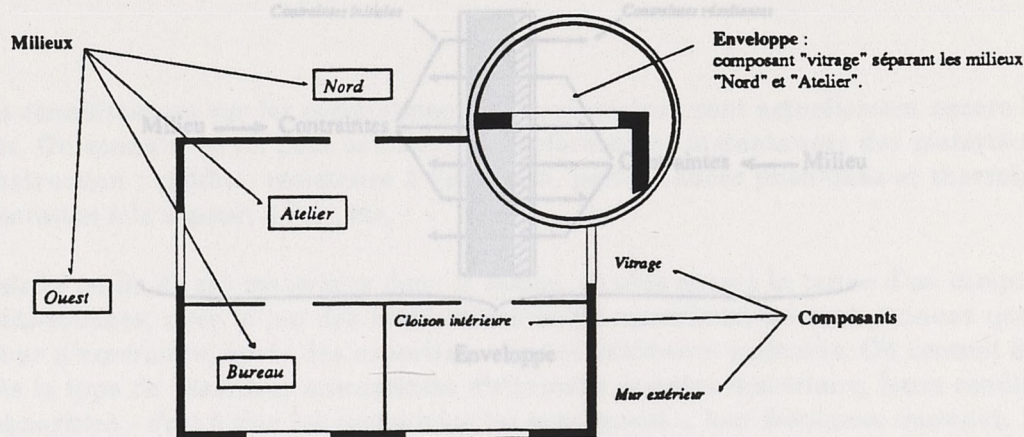
Laissant de côté tous les critères subjectifs, nous admettrons qu'une enveloppe est globalement performante si :

- elle réalise une séparation efficace entre les milieux qui l'environnent,
- elle résiste aux contraintes que produisent ces milieux, de façon à ce que sa séparation reste suffisante tout au long de sa durée de vie.

A ces deux critères principaux, nous ajouterons celui du coût de revient de l'enveloppe, ainsi que le coût de maintenance éventuel.

La terminologie que nous emploierons dans la suite de cette étude est la suivante :

- Les *milieux* définissent l'ensemble des contraintes qui sont supportées par l'enveloppe. Un milieu peut être fermé, auquel cas l'activité à l'intérieur du local est prépondérante dans la détermination des facteurs de contrainte, ou ouvert (extérieur), et prendre en compte des facteurs tels que : altitude, niveau de bruit, etc.
- Le *composant* décrit la constitution de la paroi du bâtiment : c'est un assemblage de matériaux de construction, en couches ordonnées, liées ou non (par ex : crépi / béton / laine de verre / plâtre / papier peint).
- Les *matériaux* qui composent la paroi sont les matériaux de construction que l'on peut trouver sur le marché. Ils sont qualifiés par leurs performances intrinsèques (isolation phonique, coefficients de perméabilité, résistance thermique, résilience, etc), leur épaisseur, leur coût. A remarquer que des matériaux analogues d'épaisseurs différentes seront considérés comme des matériaux différents, leurs caractéristiques ne pouvant être extrapolées à partir de leur constituant principal et de cette modification d'épaisseur (par exemple, le comportement d'une chape de béton armé ne peut être extrapolé d'après celui de blocs massifs du même matériau).
- L'*enveloppe* est le système milieu-composant-milieu, auquel on rajoutera des facteurs tels que exposition, étage, orientation, etc.
- Le *bâtiment* est composé et entouré de milieux, limité par des composants séparant ces milieux, et donc comporte autant d'enveloppes que de situations de composants.



Un bâtiment

En résumé, nous dirons que chaque couche du composant est chargée soit de remplir une fonction de séparation, soit de protéger de l'enveloppe, soit les deux à la fois. Les contraintes provenant des milieux environnants sont filtrées par ces couches, et sont propagées (dans les deux sens) en direction des autres couches du composant. La performance instantanée de l'enveloppe est fonction des contraintes résultantes après

Nous regrouperons sous le terme général de *contraintes* tous les facteurs, climatiques, d'occupation des locaux, etc, qui influent sur le composant. Ces contraintes sont de deux types : d'une part celles qui doivent être arrêtées par le composant, dans le but d'obtenir dans un milieu fermé les conditions nécessaires au bon déroulement des activités ; d'autre part, celles qui vont influencer sur le composant lui-même, en y provoquant des désordres, qui auront pour principal effet de dégrader ses performances.

Quelques exemples de contraintes, et de désordres pouvant être provoqués :

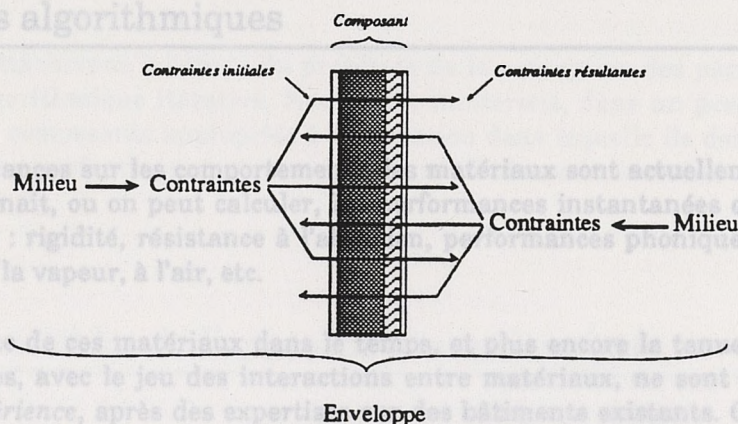
- | | |
|-----------------------------------|-------------------------------------|
| - vitesse de l'air | arrachement, défilage |
| - rayonnement UV | décoloration, dépolymérisation |
| - rayonnement IR | ramollissement, déformation |
| - contrainte structurelle | fluage, déformation |
| - humidité temporaire | gonflement, retrait hydraulique |
| - humidité et température douce | pourriture, efflorescences, mousses |
| - humidité et températures basses | gel, éclatement |
| - ... | |

Les performances globales des enveloppes sont dérivées des performances des couches du composant, c'est à dire des matériaux de construction. Comme les contraintes, elles sont de deux types : performances de séparation et de protection des autres couches du composant, performances de résistance propre. Un matériau sera considéré comme adéquat sous une contrainte s'il arrête cette contrainte (fonction de séparation), ou s'il résiste à cette contrainte (fonction de résistance).

Quelques exemples de performances des matériaux, et les contraintes arrêtées :

- | | |
|--|-------------------------------|
| - étanchéité à l'air | vitesse de l'air |
| - opacité | rayonnement UV |
| - couleur claire, résistance thermique | rayonnement IR |
| - rigidité | contrainte structurelle |
| - étanchéité à la vapeur d'eau | humidité |
| - isolation thermique | températures basses et hautes |
| - ... | |

Méthodes algorithmiques



En résumé, nous dirons que chaque couche du composant est chargée soit de remplir une fonction de *séparation*, soit une fonction de *protection* de l'enveloppe, soit les deux à la fois. Les contraintes provenant des milieux environnants sont filtrées par ces couches, et sont propagées (dans les deux sens) en direction des autres couches du composant. La *performance instantanée* de l'enveloppe est fonction des contraintes résultantes après filtrage. La *performance temporelle* de l'enveloppe est sa capacité à assurer dans le temps cette fonction de filtrage. Au fur et à mesure, sous l'action des contraintes, les couches du composant vont se dégrader, et filtrer de moins en moins les contraintes extérieures, augmentant par là même les risques de désordre, par effet boule de neige. La performance nulle est atteinte lorsqu'une couche du composant dont la fonction principale était de soutenir le tout tombe en ruine.

Problématique

Généralement, l'architecte, confronté au choix des composants du bâtiment à concevoir, retiendra de préférence des solutions éprouvées. Le nombre sans cesse croissant des matériaux de construction disponibles (surtout depuis l'apparition des composites) rend quasiment impossible l'exploration des combinaisons possibles. Des solutions ne seront jamais envisagées parce qu'elles paraissent à première vue inappropriées. De même, ne sont que trop rarement prises en compte les considérations de durabilité et de maintenance des enveloppes.

Le but de cette recherche est donc de mettre à la disposition des concepteurs un outil qui permettrait :

- d'expérimenter des solutions inusitées ; ces solutions pourraient être multiples pour autoriser un choix en fonction de critères subjectifs, à niveau de performances similaires ;
- de vérifier la validité de solutions dictées par l'utilisateur, en termes de performances de séparation instantanée ;
- de prévoir la tenue des enveloppes dans le temps ; l'analyse de leur comportement donne au concepteur un élément de choix important : est-il plus rentable de choisir un composant cher à la construction, mais plus durable, ou de choisir un composant moins onéreux, mais facilement remplaçable et à moindre coût ?

Méthodes algorithmiques

Les connaissances sur les comportements des matériaux sont actuellement encore limitées. On connaît, ou on peut calculer, les performances instantanées des matériaux de construction : rigidité, résistance à l'abrasion, performances phoniques et thermiques, séparation à la vapeur, à l'air, etc.

Mais la tenue de ces matériaux dans le temps, et plus encore la tenue d'un composant multi-couches, avec le jeu des interactions entre matériaux, ne sont connues que par *retour d'expérience*, après des expertises sur des bâtiments existants. On connaît à peu près le type de désordres susceptibles d'atteindre certains matériaux, leurs conditions d'apparition - c'est à dire les contraintes les provoquant -, leur fréquence moyenne. Mais il est très délicat de prédire le comportement global d'un composant. Tout au plus peut-on donner une fourchette de durée globale d'un composant, l'intervalle pouvant varier du simple au double suivant les experts.

Aussi il paraît très difficile de modéliser le comportement des composants, et de formaliser les lois qui permettraient, lors de la conception d'un ouvrage, de prévoir à la fois performance et durabilité de ses constituants.

L'École Polytechnique de Lausanne a mis en œuvre deux méthodes, l'une pour la conception des composants, l'autre pour l'étude des durabilités [ISE 85].

La première utilise un algorithme classique¹, qui permet, en connaissant les performances intrinsèques des matériaux de construction, de passer en revue toutes les solutions possibles, en rejetant toutes celles dont les performances instantanées globales ne sont pas satisfaisantes. C'est donc une recherche itérative multi-critères, qui consiste à trouver des couches satisfaisantes aux contraintes extérieures, et à rajouter si nécessaire des couches de séparation supplémentaires internes, en additionnant simplement les performances des matériaux constitutifs, jusqu'à ce que la performance globale du composant obtenu atteigne celle qui est attendue.

Or, dans la version actuelle, la base de données utilisée par le logiciel comprend quelques 184 matériaux répertoriés. Le nombre de couches constituant le composant, pour un bâtiment standard, est limité à 15. Même si le nombre de solutions possibles n'est pas 184¹⁵ (par exemple certains matériaux ne pouvant se trouver en face externe du composant, ou certains matériaux ne pouvant qu'être encadrés par d'autres), il est trop important pour que l'algorithme soit utilisable en situation réelle.

La méthode d'étude des durabilités, destinée avant tout à la prévision des désordres et à la maintenance du bâtiment, est aussi une méthode itérative. Connaissant le composant, sa date de construction ou de rénovation, la même base de données va servir à détecter les faiblesses des matériaux constitutifs, et à prévoir les désordres qui risquent de les affecter. Les effets de ces désordres sur leurs performances sont stockés sous forme de courbes temporelles. Elles permettent de dessiner l'évolution de l'état de la paroi et de ses performances globales. L'utilisateur peut ainsi planifier l'entretien du bâtiment.

Cette dernière méthode, qui semble prometteuse, est actuellement en cours de test en vraie grandeur.

Nous nous attacherons à l'étude du problème de la conception des parois, qui n'est pas viable en algorithmique itérative. Nous nous limiterons, dans un premier temps, à la recherche de composants appropriés à la situation dans laquelle ils doivent être placés, du point de vue performance instantanée et sensibilité aux contraintes.

Apports possibles de l'approche connexionniste

Quels sont les éléments qui empêchent l'utilisation d'algorithmes classiques ou de systèmes experts à base de règles ?

- le nombre des contraintes à intégrer : au nombre d'une cinquantaine ;
- le nombre énorme de combinaisons possibles entre les matériaux ;
- la variété des désordres pouvant affecter les matériaux, et pouvant apparaître sous la combinaison de plusieurs contraintes ;

¹ Cet algorithme est lui-même la deuxième formalisation d'une méthode qui avait été mise en œuvre dans un système expert. Cette approche a été abandonnée, dès lors que la recherche de solutions d'enveloppes pouvait se faire très méthodiquement, et qu'on pouvait se passer du côté interprétatif d'un système expert. Les deux méthodes sont aussi efficaces, la deuxième nettement plus rapide [ISE 87].

- Hypothèses
- les variations importantes des comportements des matériaux, pourtant a priori standardisés, en fonction de leur méthode de mise en œuvre, du soin apporté à leur fabrication, de contraintes particulières, qui nécessitent l'adaptation de la base de connaissance à chaque bâtiment, à chaque milieu, à chaque pays ;
 - l'apparition, à un rythme soutenu, de nouveaux matériaux ;
 - l'évolution des connaissances sur le comportement des composants, qui se fait en majeure partie par retour d'expérience.

Or, nous l'avons vu dans les chapitres précédents, les réseaux connexionnistes paraissent très efficaces pour la résolution rapide de problèmes à contraintes multiples, et très appropriés pour l'aide à la décision.

De plus, capables d'apprentissage, ils peuvent intégrer continuellement les modifications ou les ajouts apportés à la base de connaissances, ainsi que les altérations du contexte. Cette faculté, alliée à la capacité d'un réseau à généraliser, nous semble la plus importante, étant donné la masse de connaissances que devrait intégrer le système : comment formaliser le comportement de 184 matériaux, combinés à souhait, sous 50 contraintes plus ou moins liées, sans l'assistance d'un outil déductif ?

Le système idéal pour le but recherché serait un système qui, analysant les milieux environnant le composant à trouver, déterminerait les contraintes qui s'y appliquent, chercherait à la fois les matériaux les plus adéquats et leur combinaison en couches, proposerait des solutions suivant un ou plusieurs critères prioritaires (coût, durabilité, mise en œuvre, etc), et donnerait, pour chaque solution proposée, ses caractéristiques techniques (isolation thermique, phonique, poids, épaisseur, etc). La détermination des contraintes et la caractérisation des solutions retenues peut se faire par des méthodes séquentielles traditionnelles. Le cœur du système, pour la recherche de solutions optimales, serait constitué par un réseau de neurones. Ce réseau doit être conçu de façon à permettre à l'utilisateur de connaître les raisons qui font que telle ou telle solution n'a pas été retenue.

La recherche doit se dérouler en trois phases :

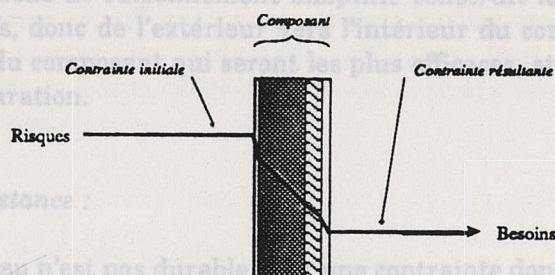
- une phase d'analyse et de formalisation d'un réseau à base de connaissances fixe et limitée, utilisé pour la validation des méthodes de recherche de solutions optimales dans un ensemble très vaste de possibles ;
- une phase d'extension des modèles proposés, afin de réaliser un réseau complet, capable d'apprentissage et d'évolution en fonction des différents contextes ;
- une phase d'intégration du réseau dans un système permettant la saisie automatique du contexte (les milieux à séparer), l'élaboration du composant, et sa validation en terme de durabilité ; son intégration dans un système de CAO existant.

Nous présenterons dans la suite de ce chapitre les travaux préliminaires réalisés. Ces travaux sont pour l'instant de simples hypothèses sur les réseaux envisagés. Nous n'avons pas pu encore faire l'acquisition d'un simulateur, et celui dont nous avons commencé l'écriture n'est pas très approprié (trop grand nombre de neurones et de connexions). Les premiers tests se feront sur la base d'un noyau logiciel en C, provenant du MIT, qu'il doit être possible de réécrire pour nos besoins.

Hypothèses

Le réseau doit proposer au concepteur une ou plusieurs solutions de composants, mono ou multi-couches. Pour ce faire, il doit résoudre un ensemble de contraintes qui peuvent être formulées comme suit :

1. *le composant doit remplir sa fonction de séparation* : d'un côté de l'enveloppe, les contraintes résultantes du filtrage - par le composant - des contraintes initiales provoquées par le milieu situé de l'autre côté ne doivent excéder un seuil déterminé par les besoins de ce milieu ; cette fonction de séparation doit s'effectuer dans les deux sens ;



Fonction de filtrage de l'enveloppe

2. *le composant doit résister aux contraintes* reçues sur ses deux faces externes ; c'est à dire que ses couches doivent être durables sous ces contraintes, initiales dans le cas des couches externes, résultantes du filtrage dans le cas des couches internes. Cette résistance doit, autant que faire se peut, être optimisée afin que les couches externes se dégradent avant les couches internes (facilité de vérification de l'état du composant, facilité de remplacement des couches).
3. *le composant doit être réalisable* : par exemple, des couches de matériaux en vrac doivent être encadrées par des couches rigides, les matériaux doivent être compatibles, la mise en œuvre doit être possible, etc.

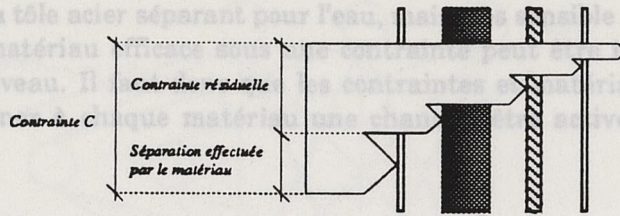
Méthode séquentielle

Si nous raisonnons en pure logique, de manière séquentielle, nous pouvons exprimer la résolution du problème par un ensemble de règles :

Règles pour la séparation :

1. Si un matériau X arrête efficacement une contrainte donnée C, choisir le matériau X ;
2. Calculer la contrainte résiduelle après filtrage de la contrainte C par X ;
3. Remonter en 1 jusqu'à ce que la contrainte résiduelle n'excède pas le niveau demandé de l'autre côté de l'enveloppe.

...y résister (cas de la tôle acier séparant pour l'eau, ... à la corrosion). De la même façon, un matériau ... fragile sous une autre, de faible niveau. Il ... matériaux interagissent en continu, pour donner un matériau ... en fonction de ses propriétés.



Comment le réseau peut-il intégrer des connaissances ? Celles-ci, en matière de compor-

Ces règles ne fonctionnent que pour une seule contrainte. Comme nous devons vérifier ces règles pour des contraintes multiples, et qu'il n'y a que peu de chances de trouver un matériau efficace pour toutes les contraintes à la fois, il faudra des opérateurs qui permettent de trouver les matériaux optimaux.

D'autre part, ce mode de raisonnement simplifié construit le composant en termes de filtrages successifs, donc de l'extérieur vers l'intérieur du composant. Ce sont donc les couches externes du composant qui seront les plus efficaces, et qui auront la plus grande utilité pour la séparation.

Règles pour la résistance :

4. Si un matériau n'est pas durable pour une contrainte donnée C, l'éliminer ;
5. Si un matériau ne peut être mis en œuvre au regard du contexte, l'éliminer.

Remarquons que, telles que formulées ci-dessus, les règles de séparation sont des règles de construction du composant, les règles de résistance des règles d'information/validation du composant obtenu.

La limitation de la méthode réside dans le fait que le composant est soumis à des contraintes sur ses deux faces ; le choix d'un matériau doit se faire en fonction des contraintes appliquées des deux côtés : or ces contraintes résiduelles ne peuvent être calculées que si l'on connaît les couches de filtrage précédentes. Il faut donc une vérification a posteriori, après constitution du composant entier. Aucun matériau n'est refusé d'entrée (si ce n'est pour sa résistance aux contraintes initiales, si on le place en face externe de composant).

Méthode parallèle

La limitation d'une méthode itérative séquentielle, nous l'avons vu, provient du très grand nombre de solutions possibles, et de la nécessité de les construire une à une, puis de les vérifier. Est-il possible, en employant des processus parallèles, d'optimiser la recherche de solutions ?

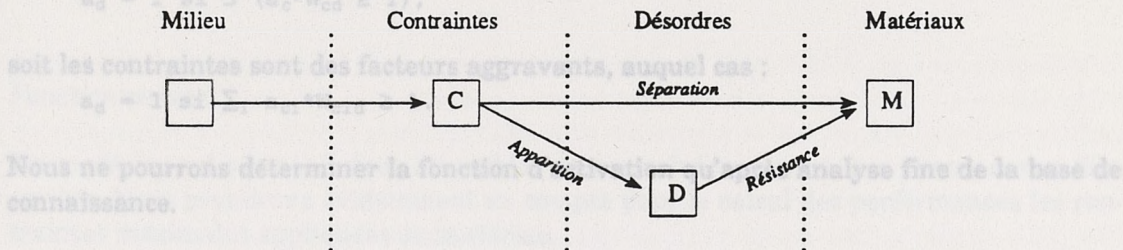
Une première analogie avec le fonctionnement des réseaux de neurones existe dans les règles 1 et 4, au niveau du choix d'un matériau seul : si on représente d'une part les contraintes, d'autre part les matériaux, comme des unités de traitement, les connexions entre contraintes et matériaux seront excitatrices si le matériau est efficace, inhibitrices si le matériau n'est pas durable, inexistantes si le matériau n'est pas concerné. Mais il faut prendre en compte qu'un matériau peut être efficace pour une contrainte mais ne pas

y résister (cas de la tôle acier séparant pour l'eau, mais très sensible à la corrosion). De la même façon, un matériau efficace sous une contrainte peut être très fragile sous une autre, de faible niveau. Il faut donc que les contraintes et matériaux interagissent en continu, pour donner à chaque matériau une chance d'être activé en fonction de ses propriétés.

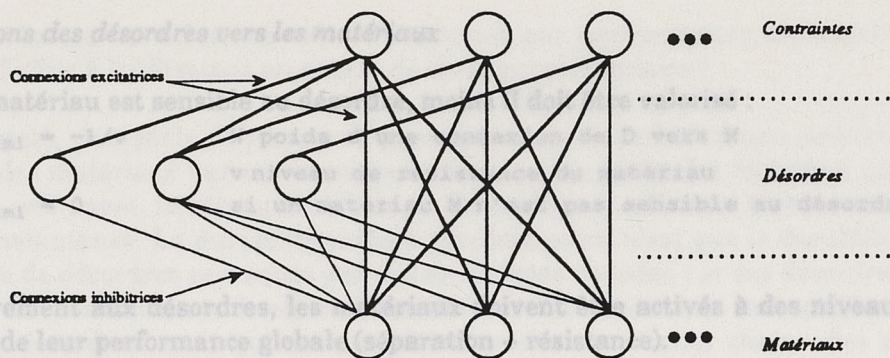
Comment le réseau peut-il intégrer des connaissances ? Celles-ci, en matière de comportement des composants du bâtiment, ne concernent que les constituants de base eux-mêmes. Les composants optimaux ne sont pas connus, ou du moins doivent être recherchés par le réseau. Le réseau ne peut donc être à apprentissage supervisé, puisque ses réponses ne sont pas évaluables en fonction de solutions connues. L'apprentissage, s'il n'est pas supervisé, ne peut consister qu'en la mémorisation des solutions optimales trouvées. Par contre, il est possible d'évaluer, pour une solution donnée, le niveau de réponse du réseau : la solution est efficace ou non, et dans quelle proportion. Nous considérerons comme bonnes toutes les solutions efficaces (qui réalisent l'adéquation contraintes/besoins), et, parmi celles-ci, le réseau évoluera vers la solution optimale, celle qui est la plus efficace.

Réseau pour le choix du meilleur matériau

En première hypothèse, nous pouvons décrire un modèle de réseau, permettant de déterminer le ou les meilleurs matériaux sous certaines contraintes, comme suit :



- trois couches d'unités de traitement sont mises en œuvre : une couche d'entrée, représentant les *contraintes*, une couche cachée *désordres*, une couche de sortie *matériaux* ;
- les contraintes sont connectées aux matériaux par une relation de type «*sépare*», le poids des connexions étant fonction de la capacité du matériau à arrêter ces contraintes ;
- les contraintes sont connectées aux désordres, par une relation de type «*provoque*», les unités de désordres étant des unités à seuil (seuil d'apparition des contraintes) ;
- les désordres sont connectés aux matériaux, par une relation de type «*altère*», le poids des connexions étant fonction de la sensibilité du matériau à ce désordre.



La connaissance que nous possédons du domaine est donc micro-codée dans les relations qu'entretiennent ces unités de traitement.

Connexions des contraintes vers les désordres

Les désordres D n'apparaissent dans les matériaux qu'à partir d'un certain niveau de contrainte C^2 . Ce seuil d'apparition est codé dans les connexions de C vers D :

$$W_{c1d1} = 1/v \quad W \text{ poids d'une connexion de C vers D}$$

$$v \text{ niveau d'apparition du désordre}$$

$$W_{c1d1} = 0 \quad \text{si un désordre D ne peut être provoqué par une contrainte C}$$

Les désordres apparaissent ou n'apparaissent pas : les niveaux d'activation des unités D sont 0 ou 1. Par contre, elles peuvent se comporter de deux façons : soit une contrainte C dépassant son seuil d'apparition provoque le désordre D :

$$a_d = 1 \text{ si } \exists (a_c * W_{cd} \geq 1),$$

soit les contraintes sont des facteurs aggravants, auquel cas :

$$a_d = 1 \text{ si } \sum_1 a_{c1} * W_{c1d} \geq 1.$$

Nous ne pourrions déterminer la fonction d'activation qu'après analyse fine de la base de connaissance.

Les connexions des contraintes C vers les matériaux M sont excitatrices - un matériau est valorisé s'il réalise au mieux la fonction de séparation à la contrainte - ; les connexions de D vers M sont inhibitrices - un matériau est dévalorisé s'il est susceptible d'être dégradé par un désordre.

Connexions des contraintes vers les matériaux

Plus le matériau est performant pour la séparation, plus il doit être valorisé :

$$W_{c1m1} = v \quad W \text{ poids d'une connexion de C vers M}$$

$$v \text{ niveau de séparation du matériau / 10 (performance maximale)}$$

2 Les niveaux de contraintes, performances, et sensibilités aux désordres sont actuellement codés sur une échelle relative, de 0 à 10 (0 absente, 1 minimum, 10 maximum). Nous utiliserons donc cette échelle pour définir les poids des connexions.

Connexions des désordres vers les matériaux

Plus le matériau est sensible au désordre, moins il doit être valorisé :

$$W_{dimi} = -1/v \quad W \text{ poids d'une connexion de D vers M}$$

v niveau de résistance du matériau

$$W_{dimi} = 0 \quad \text{si un matériau M n'est pas sensible au désordre D}$$

Contrairement aux désordres, les matériaux doivent être activés à des niveaux divers, fonction de leur performance globale (séparation + résistance).

Quel doit être leur comportement ? Si nous intégrons en entrée la notion de durabilité globale d'enveloppe, nous devons tenir compte de la sensibilité des matériaux aux désordres : un composant dont on exige une durabilité illimitée ne peut supporter aucun désordre, un composant à faible durabilité peut admettre des désordres prononcés sur ses matériaux. La durabilité à atteindre, variable globale d'entrée du système, doit être utilisée pour accepter ou rejeter un matériau. Cette durabilité n'ayant d'effet que vis-à-vis des désordres, on doit traiter différemment entrées inhibitrices et excitatrices :

$$\text{inhibition} = \sum_i a_{di} * W_{dim}$$

$$a_m = 0 \quad \text{si } (|\text{inhibition}|/n > \text{durabilité})$$

$$a_m = \sum_i a_{ci} * W_{cim} + \text{inhibition} \quad \text{dans le cas contraire.}$$

où n nombre d'entrées inhibitrices (désordres)
 a_m activité du matériau m

la durabilité est exprimée dans $[0..1]$

Fonctionnement

Un matériau est soumis à deux ensembles de contraintes : sur sa face avant, sur sa face arrière. Nous prendrons évidemment en compte pour le calcul des performances les contraintes maximales appliquées au matériau.

La définition du réseau doit lui permettre de choisir, en une seule passe, le ou les matériaux candidats (plusieurs matériaux peuvent présenter les mêmes performances). Une passe complète consiste à :

- activer les contraintes à leur niveau, déterminé par les milieux ;
- propager vers les désordres, afin d'activer ceux susceptibles d'être provoqués ;
- propager des contraintes et des désordres vers les matériaux.

Limitations

Ce réseau fonctionne dans le seul but de trouver le meilleur matériau, c'est à dire le plus séparant, et le plus durable. Par contre, d'une part il peut ne pas suffire à assurer les besoins de séparation attendus, d'autre part il peut ne pas être optimal dans la mesure où sa fonction de séparation et de résistance peut dépasser les besoins fixés. Devons-nous rechercher la meilleure solution - si tant est qu'elle existe -, ou devons-nous arrêter la recherche dès lors qu'une solution globalement satisfaisante est atteinte ? Devons-nous

attendre toutes les solutions valides³, classées par performances croissantes, ce qui permet d'offrir à l'utilisateur un critère de choix supplémentaire ?

D'autre part, ce modèle de réseau ne prend pas en compte le fait que certaines performances des matériaux varient au cours du temps, sous l'effet des désordres qu'ils subissent. En une passe, le réseau choisit le ou les matériaux efficaces en termes de performance instantanée. La durabilité utilisée précédemment n'est que la durabilité sous un ensemble de désordres provoqués par des contraintes données ; or ces désordres peuvent affaiblir des performances de séparation, pour des contraintes non susceptibles de provoquer des désordres, mais qui auraient fait rejeter - en premier choix - des matériaux inaptes à les assurer efficacement.

Une solution pourra être étudiée, qui permet une rétro-action des désordres sur les performances de séparation et de sensibilité des matériaux. Le niveau d'activation d'un matériau devra prendre en compte ces phénomènes, ainsi que son activité antérieure (pourquoi il a été choisi en premier lieu). Une autre solution peut consister en l'intégration d'une valeur de «gravité» du désordre sur le matériau, qui désactiverait les matériaux en fonction de la vitesse de la perte de performance globale sous ce désordre. Il est auparavant nécessaire de valider le comportement du réseau tel que défini ci-dessus, afin de savoir si son niveau de définition est suffisant ou non.

Nous nous limiterons, dans un premier temps - et pour ne pas augmenter inconsidérément le nombre de paramètres de choix - à l'étude des performances instantanées, ne prenant pas en compte l'évolution possible de ces performances dans le temps.

Réseau pour la construction du composant

Le modèle de réseau décrit ci-dessus peut permettre de déterminer, sous un ensemble de contraintes considérées comme entrées du système, le meilleur matériau. La construction du composant, c'est à dire le choix de différents matériaux et leur organisation en couches ordonnées, relève d'un autre modèle.

Pour le but recherché, il est impossible de sélectionner en un seul passage, pour une couche donnée, un matériau : ces couches sont en perpétuelle interaction, dans la mesure où chacune d'entre elles va filtrer les contraintes qu'elle reçoit. Ainsi, le choix d'un matériau d'une couche, abaisse le niveau des contraintes appliquées sur les couches adjacentes, et vice-versa, ce qui abaisse donc le niveau de performances demandé à ces couches.

D'autre part, sachant qu'il est rare de trouver des solutions efficaces à une couche, il est nécessaire, dans ce modèle, de prendre en compte la notion de réalisation des besoins qui n'était pas utilisée dans le modèle précédent : le réseau doit d'abord expérimenter les solutions à une couche, trouver le meilleur matériau, puis, si les besoins ne sont pas résolus, rajouter des couches. Une bonne solution est celle qui, à un moment donné de la recherche, satisfait les besoins ; les solutions ultérieures ne seront que des améliorations apportées en termes de performance globale ; la solution optimale sera atteinte en fin de recherche.

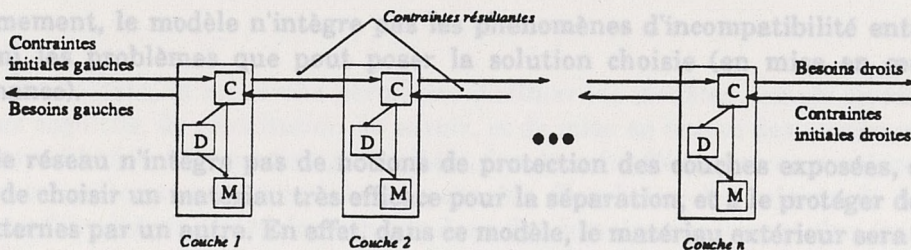
³ Le coût du composant étant une contrainte d'entrée du système, ces solutions valides doivent forcément aussi satisfaire cette contrainte ; elles doivent donc être de niveau de coût équivalent.

Enfin, le réseau doit ordonner les couches sélectionnées : la répartition des contraintes sur les couches n'est pas symétrique, les niveaux initiaux et les filtrages étant différents. La réponse globale du système (le composant optimal) ne peut consister uniquement en un ensemble de micro-solutions (le matériau optimal pour une situation donnée). Cette réponse doit intégrer également l'ordonnement de ces micro-solutions. Le réseau doit être conçu comme un système dont la réponse est l'ordre des réponses de sous-systèmes chargés chacun d'une couche particulière et repérable.

Description

Le réseau est constitué d'une chaîne de sous-réseaux du type décrit précédemment, et représentant les couches du composant. Si nous prenons comme convention de numéroter ces couches de gauche à droite, de 1 à n (n impair), les contraintes appliquées à gauche du composant sont propagées - et filtrées - vers la droite, de la couche n à la couche n + 1, et les contraintes appliquées à droite sont propagées de la couche n à la couche n - 1.

Une face d'une couche n est donc soumise soit aux contraintes initiales, si elle est externe au composant, soit aux contraintes résultantes du filtrage par les couches qui la protègent. Une couche n est soumise au maximum des contraintes rencontrées à gauche et à droite.



Fonctionnement

Exemple de fonctionnement possible (quasiment idéal) :

- Les contraintes initiales sont propagées au cœur du système ;
- Le réseau central détermine le matériau le plus approprié ;
- Les contraintes résultantes sont calculées (à droite et à gauche) ;
- Si elles sont supérieures aux besoins du côté correspondant, la solution monocouche n'est pas satisfaisante, et une couche doit être rajoutée de ce côté ;
- La recherche d'une deuxième couche est enclenchée ;
- Le matériau sélectionné pour cette deuxième couche filtre les contraintes initiales qu'il reçoit de son côté, les envoie vers la couche déjà déterminée ;
- Ces nouvelles contraintes, appliquées à la première couche, font sélectionner un matériau de niveau de performance plus faible ;
- Ce nouveau matériau filtre différemment les contraintes qu'il reçoit, et le processus reprend dans l'autre sens, les deux couches interagissant jusqu'à stabilisation ;
- Si les besoins ne sont toujours pas satisfaits, on reprend en augmentant le nombre de couches.

Ce modèle nécessite un détecteur de stabilité, qui permet de savoir quand une solution à

m couches est optimale, et un organe de décision calculant le niveau de réponse obtenue, et si la solution n'est pas suffisamment efficace, décide qu'il faut introduire s'il faut rechercher une solution à $m+1$ couches.

Limitations

Ce modèle comporte, si l'on se place au niveau théorique, des lacunes importantes. Nous listerons ci-dessous certaines d'entre elles, tout en sachant qu'elles ne permettront de rejeter le modèle que s'il s'avère, lors des tests, qu'elles sont rédhibitoires.

Première lacune, le fait que les contraintes résiduelles d'un côté du composant sont utilisées pour déterminer qu'il faut rajouter une couche de ce côté. Le modèle fonctionne donc à première vue en termes de filtrages successifs, ce qui est peut-être dommageable pour la recherche de solutions inusitées. Les contraintes résiduelles doivent-elles être réinjectées, d'une manière ou d'une autre, en amont de la couche inefficace ?

Deuxième lacune, le mode d'interaction entre les couches adjacentes. S'il est nécessaire de rajouter une couche en aval d'une déjà déterminée, qui donc la protégera, on peut choisir pour la première un matériau moins efficace. Il faut donc soit recommencer le calcul à zéro (ne pas tenir compte du matériau précédemment choisi), soit diminuer le niveau d'activation de celui-ci, de façon à autoriser l'émergence d'une nouvelle solution.

Troisièmement, le modèle n'intègre pas les phénomènes d'incompatibilité entre matériaux, ni les problèmes que peut poser la solution choisie (en mise en œuvre, en maintenance).

Enfin, le réseau n'intègre pas de notions de protection des couches exposées, celle qui permet de choisir un matériau très efficace pour la séparation, et à le protéger des agressions externes par un autre. En effet, dans ce modèle, le matériau extérieur sera celui qui recevra les contraintes maximales ; il est donc choisi de façon à minimiser ses désordres éventuels et à la fois à maximiser ses pouvoirs séparants. Si l'on réinjecte les désordres en amont, d'une manière ou d'une autre, de façon à déclencher la recherche d'une couche le protégeant, celle-ci sera à son tour soumise aux mêmes contraintes initiales. Tel quel, le réseau pour cette couche trouvera le même matériau que précédemment. Le réseau global déplace les couches, et ne peut trouver de solution. Faut-il encoder explicitement cette fonction de protection dans le système, au risque d'obtenir un réseau plus volumineux et plus lent qu'un algorithme séquentiel ? Faut-il raisonner en termes de performance globale, et sélectionner les couches non plus seulement en fonction des contraintes, mais en incluant la performance déjà atteinte avec les matériaux choisis ?

Conclusions

On le voit, les limitations théoriques du modèle, du moins dans le premier état envisagé, proviennent de points obscurs dans le mode de validation des hypothèses de solutions. La construction d'une solution de composant met en jeu des notions diverses et contradictoires, notion de filtrages successifs, notion de protection, de résistance, de sensibilité instantanée, de durabilité, etc. Il est nécessaire, avant de poursuivre l'élaboration d'un modèle de réseau, de tester les différentes approches sur des cas réels, pour estimer la validité des approches choisies. Comme dans la plupart des modèles de réseau construits à ce jour, la conception d'un modèle se fait par expérimentations successives, avec une bonne dose d'intuition et d'empirisme.

Bibliographie

L'avantage de l'étude décrite dans le chapitre précédent réside dans la possibilité d'intégrer les outils obtenus dans des logiciels de CAO existants, et par là même, par des tests en vraie grandeur, d'entériner ou d'invalidier la méthode retenue. Si celle-ci s'avère satisfaisante, tant sur le plan de l'efficacité que de la rapidité, il est possible d'envisager son extension à d'autres domaines, par exemple la conception de structures, l'implantation de locaux ou d'ateliers, etc.

[HEB 49] D.O. Hebb, "The Organization of Behavior". New York : Wiley, 1949.

A plus long terme, l'objectif est de construire un modèle générique d'outils d'aide à la décision. Ce modèle permettrait l'acquisition de connaissances et leur évolution, la définition des contraintes de résolution du problème, la récupération des propositions de solutions élaborées par le système. De plus, ce système devra, autant que possible, être capable d'explicitier les raisons de ses choix.

Ce modèle pourra être décliné sous forme de composants logiciels prêts à l'intégration dans les logiciels de conception assistée, porteurs de connaissances ou non, capables d'apprentissage et d'adaptation, pour tous les domaines où la prise de décision ne peut se faire de façon algorithmique classique, au regard du nombre de contraintes à prendre en compte.

Fondamentalement, la recherche permettra d'affiner les modèles actuels en matière de processus cognitifs, de constitution du savoir, et de mise en œuvre des savoirs et savoir-faire. Les méthodes d'apprentissage étudiées, la nécessité d'élaborer un système a priori basé sur un raisonnement intuitif, mais capable d'explicitier les raisons de ses choix, pourraient faire l'objet de développements dans des domaines beaucoup plus étendus que la conception assistée.

[LON 89] J. Lencelle, "Détection de contours par rétropropagation du gradient", Neuro-Nîmes, 1989.

[MIN 69] M. Minsky, & S. Papert, "Perceptrons". Cambridge, MA : MIT Press, 1969.

[MKE 86] Mc Keown, F. Pate, "Alignment and connection of fragmented linear features in aerial imagery", IEEE, 1985.

[RUM 86] D. Rumelhart, G. Hinton, J. McClelland, "A general Framework for Parallel Distributed Processing", *Parallel Distributed Processing*, MIT Press, 1986.

[SER 86] J. Serra, "Mathematical Morphology", CVIP N° 35, 1986.

[SUT 81] Sutton & Barto, "Toward a modern theory of adaptive networks : Expectation and prediction". *Psychological Review*, 88, pp 135-170, 1981.

[TAG 89] P. Tagliarino, "De l'étude de la transformée de Hough", Thèse de Doctorat, Université de Saint-Etienne, 1989.

[THO 89] Thomson Sintra ASM (Arcueil, F), "Identification automatique de bruits impulsifs en acoustique sous-marine par réseaux-multicouche", Neuro-Nîmes, 1989.

Bibliographie

- [DAV 90] E.R. Davis, *"Machine Vision"*, Academic Press, 1990.
- [HEB 49] D.O. Hebb, *"The Organization of Behavior"*. New York : Wiley, 1949.
- [HIN 81] G.E. Hinton, "Implementing semantic networks in parallel hardware". In G.E. Hinton & J.A. Anderson (Eds), *Parallel models of associative memory*, pp. 161-188, Hillsdale, NJ : Erlbaum, 1981.
- [HIN 84] G.E. Hinton, T.J. Sejnowski, D.H. Ackley, "Boltzmann machines : constraint satisfaction networks that learn", *Technical report n° CMU-CS-84-119*, Carnegie-Mellon University, Pittsburgh, 1984.
- [HOF 79] D.R. Hofstadter, *"Gödel, Escher, Bach : an Eternal Golden Braid"*. New York : Basic Books, 1979.
- [ISE 85] F. Iselin, J. Menu, L. De Trintignan, "ACE : une Aide à la Conception des Enveloppes architecturales", *5ème Journée Internationale sur les Systèmes Experts et leurs Applications*, Avignon, mai 1985.
- [ISE 87] F. Iselin, J. Menu, "D'un système expert à un algorithme : ACE-3", *Revue d'Intelligence Artificielle*, 1, pp 55/74, Hermès, Paris, 2.1987.
- [KIR 83] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, "Optimization by simulated annealing", *Science* v.220, 1983.
- [KOH 84] T. Kohonen, *"Self-organization and associative memory"*, Springer Verlag, 1984.
- [LON 89] J. Loncelle, "Détection de contours par rétropropagation du gradient", Neuro-Nîmes, 1989.
- [MIN 69] M. Minsky, & S. Papert, *"Perceptrons"*. Cambridge, MA : MIT Press, 1969.
- [MKE 85] Mc Keown, F. Pane, "Alignment and connection of fragmented linear features in aerial imagery", IEEE, 1985.
- [RUM 86] D. Rumelhart, G. Hinton, J. McClelland, "A general Framework for Parallel Distributed Processing", *Parallel Distributed Processing*, MIT Press, 1986.
- [SER 86] J. Serra, *"Mathematical Morphology"*, CVIP N° 35, 1986.
- [SUT 81] Sutton & Barto, "Toward a modern theory of adaptative networks : Expectation and prediction". *Psychological Review*, 88, pp 135-170, 1981.
- [TAG 89] P. Tagliarino, *"De l'étude de la transformée de Hough"*, Thèse de Doctorat, Université de Saint-Etienne, 1989.
- [THO 89] Thomson Sintra ASM (Arcueil, F), "Identification automatique de bruits impulsifs en acoustique sous-marine par réseaux-multicouche", Neuro-Nîmes, 1989.

image 1

Annexe : Analyse d'images

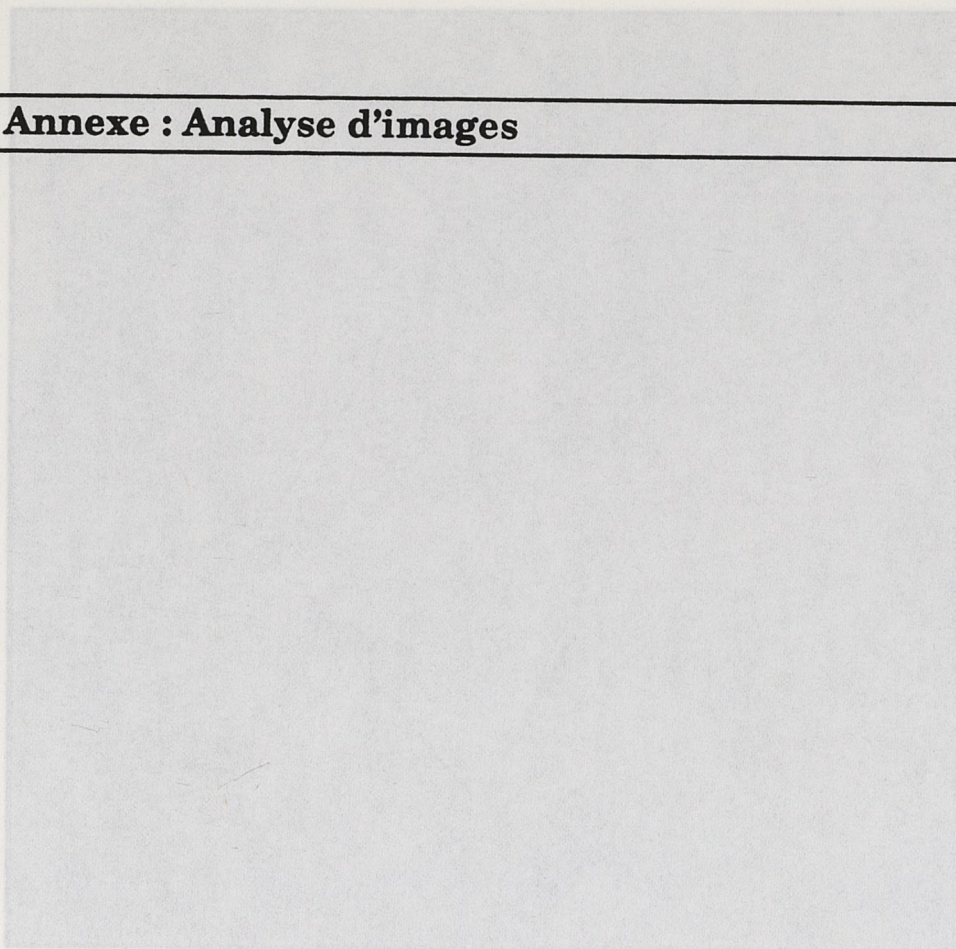


image 2



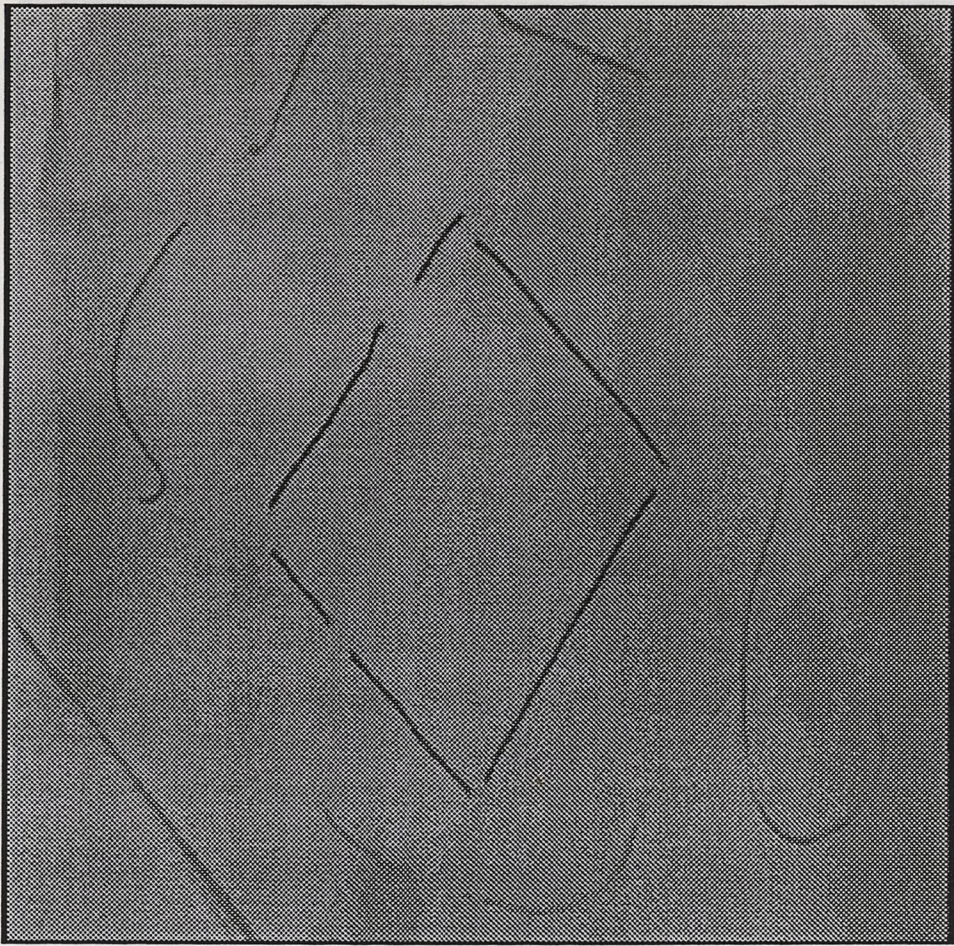


image 1

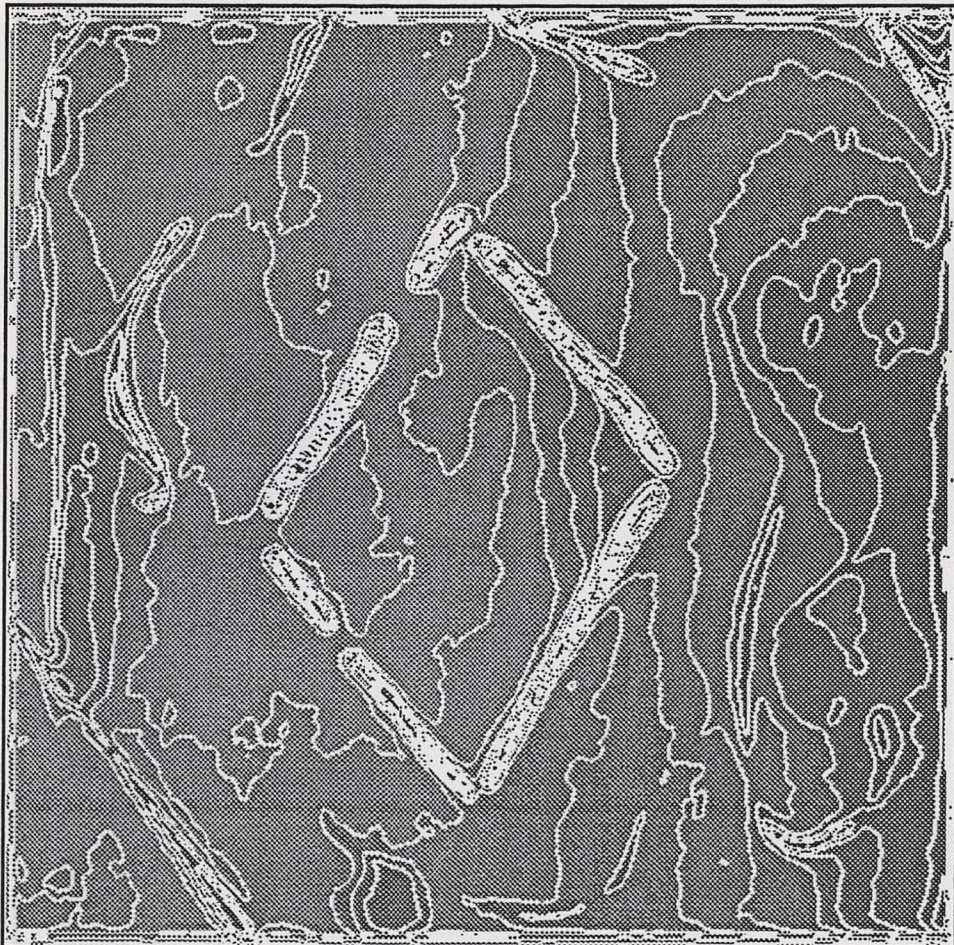


image 2



image 3

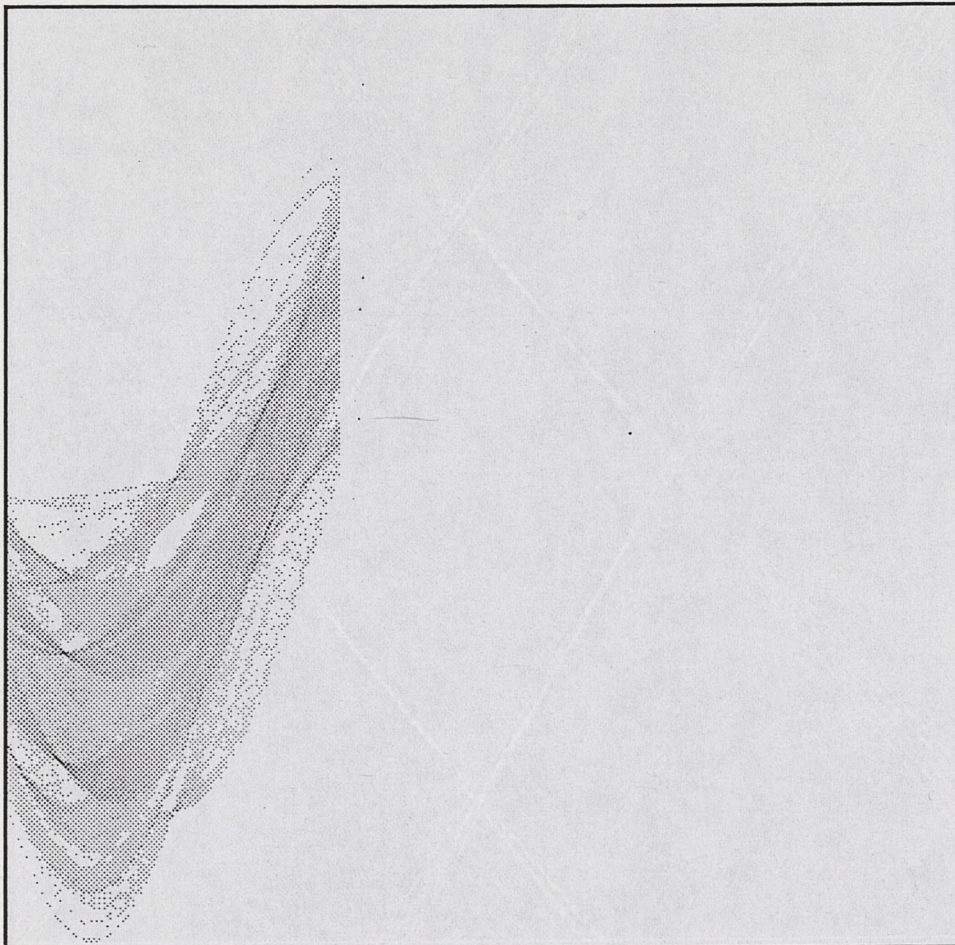


image 4 & 1
superposées

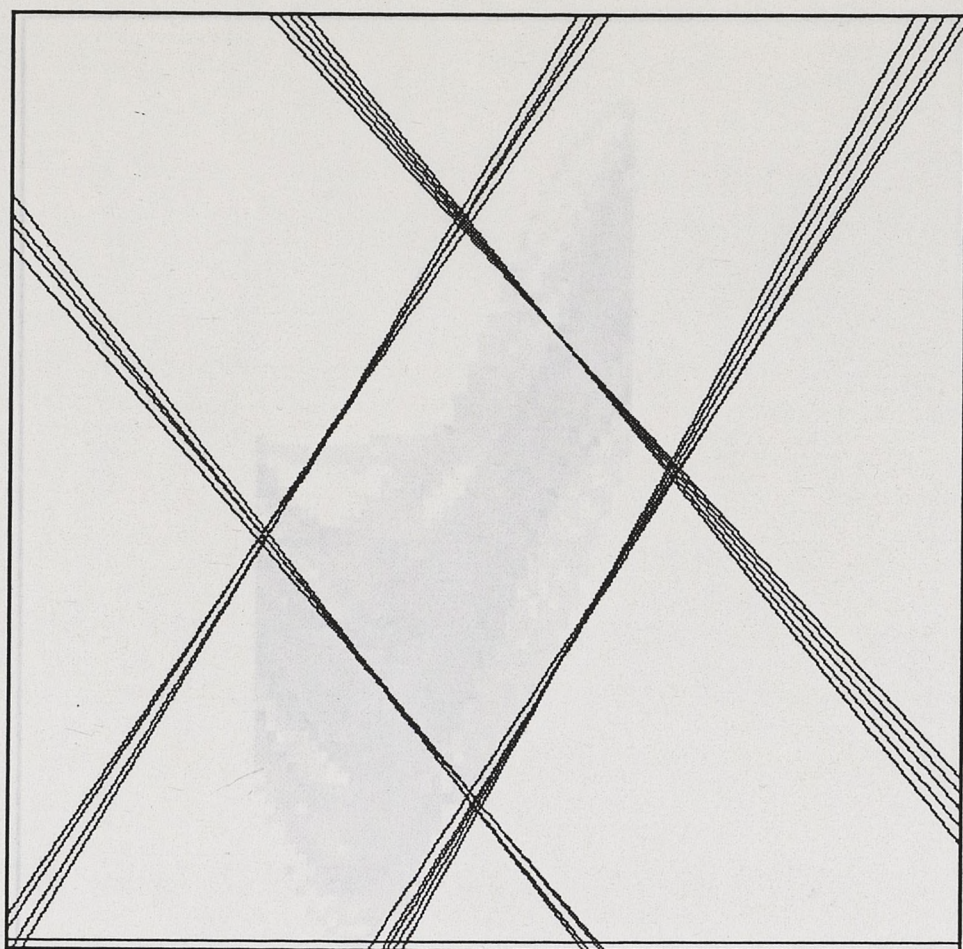
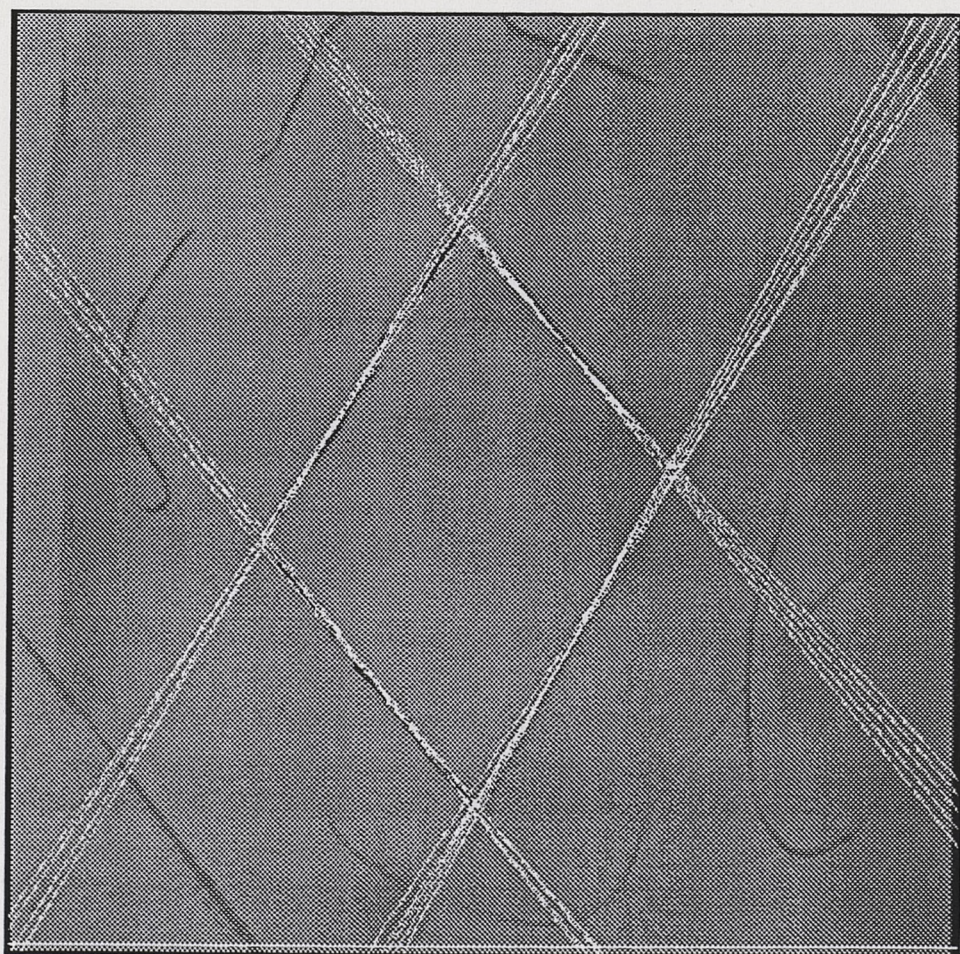
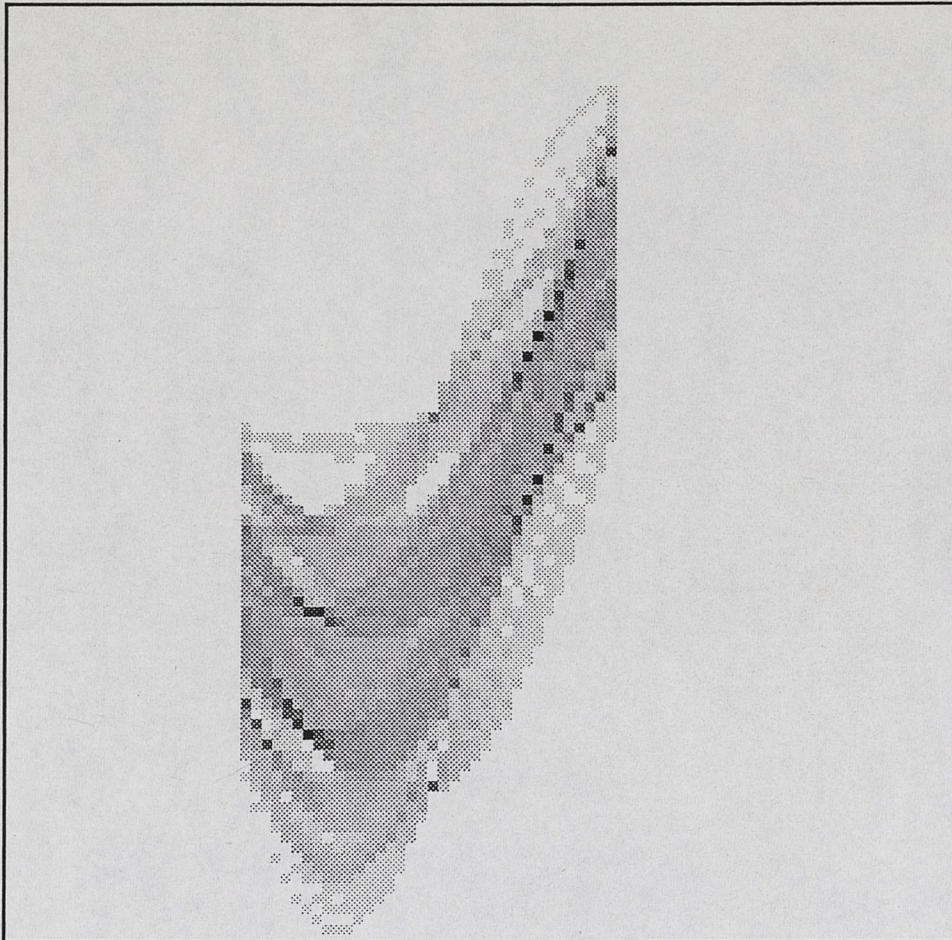


image 5



images 5 & 1
superposées
superposées à 1

image 6



inverse de
l'image 6
superposées à 1

