Réseau de neurones dynamique perceptif - Application à la reconnaissance de structures logiques de documents

Yves Rangoni

Nancy-Université
Université Nancy 2

École doctorale IAEM Lorraine

UFR mathématiques et informatique

Département de formation doctorale en informatique

Réseau de neurones dynamique perceptif – Application à la reconnaissance de structures logiques de documents

THÈSE

présentée et soutenue publiquement le 9 novembre 2007

pour l'obtention du

Doctorat de l'université Nancy 2

(spécialité informatique)

par

Yves Rangoni

Composition du jury

Président :	Anne Boyer	Professeur, Université Nancy 2
Rapporteurs :	Rolf Ingold	Professeur, Université de Fribourg, Suisse
	Thierry Paquet	Professeur, Université de Rouen
Examinateurs :	Hubert Emptoz	Professeur, INSA Lyon
	Abdel Belaïd	Professeur, Université Nancy 2

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Loria

Mis en page avec la classe thloria.

Nancy-Université
Université Nancy 2

École doctorale IAEM Lorraine

UFR mathématiques et informatique

Département de formation doctorale en informatique

Réseau de neurones dynamique perceptif – Application à la reconnaissance de structures logiques de documents

THÈSE

présentée et soutenue publiquement le 9 novembre 2007

pour l'obtention du

Doctorat de l'université Nancy 2

(spécialité informatique)

par

Yves Rangoni

Composition du jury

Président :	Anne Boyer	Professeur, Université Nancy 2
Rapporteurs :	Rolf Ingold	Professeur, Université de Fribourg, Suisse
	Thierry Paquet	Professeur, Université de Rouen
Examinateurs :	Hubert Emptoz	Professeur, INSA Lyon
	Abdel Belaïd	Professeur, Université Nancy 2

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Loria

Mis en page avec la classe thloria.

Résumé

L'extraction de structures logiques de documents est un défi du fait de leur complexité inhérente et du fossé existant entre les observations extraites de l'image et leur interprétation logique. La majorité des approches proposées par la littérature sont dirigées par le modèle et ne proposent pas de solution générique pour des documents complexes et bruités. Il n'y a pas de modélisation ni d'explication sur les liens permettant de mettre en relation les blocs physiques et les étiquettes logiques correspondantes. L'objectif de la thèse est de développer une méthode hybride, à la fois dirigée par les données et par le modèle appris, capable d'apprentissage et de simuler la perception humaine pour effectuer la tâche de reconnaissance logique. Nous avons proposé le Réseau de Neurones Dynamique Perceptif qui permet de s'affranchir des principales limitations rencontrées dans les précédentes approches. Quatre points principaux ont été développés :

- utilisation d'une architecture neuronale basée sur une représentation locale permettant d'intégrer de la connaissance à l'intérieur du réseau. La décomposition de l'interprétation est dépliée à travers les couches du réseau et un apprentissage a été proposé pour déterminer l'intensité des liaisons ;
- des cycles perceptifs, composés de processus ascendants et descendants, accomplissent la reconnaissance. Le réseau est capable de générer des hypothèses, de les valider et de détecter les formes ambiguës. Un retour de contexte est utilisé pour corriger les entrées et améliorer la reconnaissance ;
- un partitionnement de l'espace d'entrée accélérant la reconnaissance. Des sous-ensembles de variables sont créés automatiquement pour alimenter progressivement le réseau afin d'adapter la quantité de travail à fournir en fonction de la complexité de la forme à reconnaître ;
- l'intégration de la composante temporelle dans le réseau permettant l'intégration de l'information de correction pendant l'apprentissage afin de réaliser une reconnaissance plus adéquate. L'utilisation d'un réseau à décalage temporel permet de tenir compte de la variation des entrées après chaque cycle perceptif tout en ayant un fonctionnement très proche de la version statique.

Mots-clés : analyse d'images de documents, réseau de neurones transparent, Perceptron multicouche, réseau de neurones dynamique, cycle perceptif, correction et sélection de variables

Abstract

Logical structure extraction of documents remains a challenging problem due to their inherent complexity and the gap between the physical features extracted from the image and their corresponding logical interpretation. Most of the literature approaches propose model-driven approaches which are not generic enough to handle complex and noisy documents. They do not use intermediate interpretation steps and do not explain the relationships between the physical blocks and the corresponding logical labels. The main objective of this thesis is to develop a hybrid method, using both data-driven and model-driven approach, which is capable to learn the relationships and simulate human perception during the logical recognition task. We have proposed a Dynamic Perceptive Neural Network which can handle drawbacks of previous systems. Four main points have been developed:

- a special network topology based on local representation where the knowledge can be integrated. The logical interpretation is unfolded along the layers of the network and a training stage is performed to find the weights for each link;
- perceptive cycles (several bottom-up and top-down processes) perform the recognition. The network is able to generate hypothesis, validate them and detect ambiguous patterns. The context manages the correction of the input features to improve the recognition rate;
- an input feature clustering has been proposed to speed-up the recognition. Subsets of features are automatically computed and are given progressively to feed the network in order to adapt the amount of computations according to the pattern complexity;
- dynamic integration in the network that make it possible to integrate the data correction information during the training stage to have more appropriate behavior during the recognition. The improvement uses a Time Delay Neural Network architecture to take into account the input data variations after each perceptive cycle while the recognition step is quite similar to the static one.

Keywords: document image analysis, transparent neural network, multilayer Perceptron, dynamic neural network, perceptive cycle, feature correction and selection

Remerciements

Je tiens en premier lieu à remercier le Professeur Abdel Belaïd qui m'a permis de commencer cette thèse sous sa direction et de son soutien humain et scientifique pour mener à bien ce travail durant ces quatre dernières années.

Je souhaite remercier les Professeurs qui m'ont fait l'honneur de participer à mon jury de thèse, les professeurs Rolf Ingold de l'Université de Fribourg et Thierry Paquet de l'Université de Rouen, qui ont eu la difficile tâche d'être les rapporteurs de ce mémoire de thèse, le Professeur Hubert Emptoz en tant qu'examinateur et le Professeur Anne Boyer en tant qu'examinateur interne. Merci donc pour l'attention qu'ils ont apportée à ce travail de thèse ainsi que pour leur remarques et critiques constructives.

Je souhaite également remercier tous les membres de l'équipe READ pour leur accueil, leur soutien et leur aide qui m'ont permis d'évoluer dans un cadre scientifiquement dynamique et dans une ambiance de travail plus qu'agréable. Merci en particulier à Yolande Belaïd pour son aide constante tout au long de ma thèse, à Christophe pour m'avoir aidé dans les premières phases de l'élaboration du réseau de neurones perceptif, Hubert et Szilard pour avoir partagé avec moi leurs connaissances sur le PMC et sur bien d'autres sujets, André pour le projet Paploo, le trop court passage de Jacques dans notre bureau, Hatem pour m'avoir épaué ces derniers mois. Bonne chance à lui qui termine d'ici peu et à Nazih, monsieur Matlab, au milieu du parcours. Merci à tous les stagiaires pour leur aide, en particulier Nicolas et Jonathan pour leur contribution lors de la conception de l'interface d'étiquetage.

Un grand merci à tous les autres membres du LORIA avec qui j'ai pu partager de bons moments de travail ou de détente, les anciens camarades de feu l'équipe ISA, Fabien, Rodrigo, Gilles, Sébastien et les nouveaux VEGAS, Julien et Luiz, les Orpailleur Laszlo, Rokia, Nizar, Adrien. Chedia et Stéphane de l'équipe SITE, Benjamin l'inclassable et tous ceux que je n'ai pas cités comme les Qgar ou les expatriés du bâtiment C, mais que je remercierai au pot. Une pensée aux anciens messins : Pierre, Alain, Nicolas, à tous les moniteurs que j'ai rencontrés au LORIA ou lors de nos fameux ateliers : Adnene, Mohamed, Salah, Alan et les membres de l'atelier Fête de la Science, les membres de l'atelier Science et éthique. Merci aussi à ceux qui m'ont aiguillé pendant ce monitorat, Azim Roussanaly, Antoine Tabbone, Daniel Coulon et bien sûr toute l'équipe pédagogique de l'UFR MI, trop nombreux à citer. Un merci bien sûr à Jacques Lonchamp et Nadia Bellalem qui m'ont encadré lors de mon ATER ainsi que toute l'équipe pédagogique de l'IUT Charlemagne (elle aussi trop nombreuse à citer). Merci à ceux dont l'enseignement m'a permis d'arriver au grade de docteur et aussi aux étudiants qui m'ont eu comme enseignant et qui ont aussi contribué indirectement à mon accession à ce stade.

Je tiens également également à remercier mes parents, mon frère, ma compagne Sylvie et mon fils Téo pour m'avoir toujours soutenu et à qui je dois tout.

Je suis sûr d'avoir oublié une quantité incroyable de personnes par fuite de mémoire temporaire ou par faute de place, mais ceux qui me connaissent savent qu'il peuvent compter sur moi bien même après mon départ du LORIA.

Yves Rangoni

À mon fils Téo, à Sylvie sa maman
à toute ma famille

Table des matières

Introduction **1**

Chapitre 1
Analyse et reconnaissance d'images de documents **5**

1.1 Définitions 5
1.2 Reconnaissance de documents 6
1.3 Structure physique, structure logique 8
1.4 Méthodologies pour la reconnaissance de documents 11
1.4.1 Approches dirigées par le modèle 11
1.4.2 Approches dirigées par les données 20
1.4.3 Approches descendantes et ascendantes 22
1.5 Évaluation des performances 24
1.6 Discussion des méthodes 25
1.7 Conclusion 28

Chapitre 2
Réseaux de neurones à représentation locale et utilisation du contexte **31**

2.1 Introduction 31
2.2 Modèles cognitifs de lecture 32
2.3 Systèmes de lecture basés sur des principes cognitifs 33
2.3.1 Le système Perceptro 34
2.3.2 Réseau de neurones transparent 42
2.4 Le Perceptron multicouche 43
2.4.1 Le neurone 43
2.4.2 Topologie en couches 44
2.4.3 Apprentissage 46
2.4.4 Applications 49
2.5 Conclusion 50

Chapitre 3
Réseau de neurones perceptif **53**

3.1 Système proposé 53
3.1.1 Choix des primitives 54
3.1.2 Structures logiques 56

3.1.3 Contexte 56
3.1.4 Apprentissage 59
3.2 Reconnaissance par cycles perceptifs 61
3.2.1 Propagation 61
3.2.2 Analyse 62
3.2.3 Correction 63
3.2.4 Cycles perceptifs 65
3.3 Expérimentations 67
3.4 Conclusion 71

Chapitre 4
Méthode de partitionnement **73**

4.1 Réseau de neurones perceptif et temps de reconnaissance 73
4.2 Accélération de la reconnaissance 74
4.3 Méthodes diminuant la taille de l'entrée 76
4.3.1 La sélection de variables 76
4.3.2 Classement de variables 76
4.3.3 Sélection de sous-ensembles de variables 78
4.3.4 Réduction de données 80
4.4 Partitionnement de l'espace d'entrée 83
4.4.1 Contraintes sur le choix de la méthode à proposer 83
4.4.2 Justification de la méthode 85
4.4.3 Algorithme de la méthode 87
4.4.4 Choix de la dimension du sous-espace 91
4.5 Expérimentations 93
4.6 Conclusion 100

Chapitre 5
Réseau de neurones dynamique perceptif **103**

5.1 Réseau de neurones perceptif et correction des entrées 103
5.2 Réseaux dynamiques 104
5.2.1 Réseaux statiques récurrents 104
5.2.2 Autres réseaux dynamiques 106
5.2.3 Difficultés des réseaux dynamiques 107
5.2.4 Choix du réseau 107
5.3 Réseau à décalage temporel 108
5.3.1 Topologie et fonction d'activation 108
5.3.2 Apprentissage 109
5.4 Réseau de neurones dynamique perceptif 112
5.5 Expérimentations 115
5.6 Perspectives 117
5.7 Conclusion 121

Conclusion et perspectives **123**

Annexes

Annexe A
La base des articles scientifiques **129**

Annexe B
La base MNIST **139**

Annexe C
Perceptron multicouche

C.1 Rétropropagation du gradient de l'erreur 143
C.2 Propriétés mathématiques 145

Annexe D
Réseaux récurrents statiques

D.1 Descente de gradient 149

Annexe E
Détermination des valeurs et vecteurs propres **153**

E.1 Méthodes du polynôme caractéristique 153
E.2 Méthode de la puissance itérée 153
E.3 Méthodes des matrices semblables 154
E.3.1 Méthode de Crout 154
E.3.2 Méthode de Rutishauser 155
E.3.3 Méthode QR 155

Bibliographie **159**

Index des auteurs **175**

Index **181**

Table des figures

1.1 Image de document, décomposition physique et logique en arbre . . . 8
1.2 Représentation hiérarchique des structures de documents . . . 9
1.3 Composition générale d'un fichier TEI . . . 10
1.4 Instance de structure logique pour un article scientifique selon Summers . . . 11
1.5 Exemple de règles pour la détection du titre par Kim et coll. . . . 12
1.6 Exemple de règles utilisées par Niyogi et Srihari . . . 14
1.7 Exemple de modélisation dans Graphein d'une page d'un article de revue par un modèle physico-logique générique . . . 15
1.8 Exemple de grammaire utilisée par Conway . . . 17

2.1 Représentation d'un système humain de traitement de l'information appliqué au processus de lecture selon Baccino et coll. . . . 33
2.2 Mot manuscrit cursif et problème de segmentation . . . 34
2.3 Observations expérimentales de la supériorité du mot sur la lettre isolée . . . 36
2.4 Modèle d'activation interactive . . . 36
2.5 Types de connexions entre les niveaux et les neurones selon McClelland et Rumelhart . . . 37
2.6 Propagation de l'activation chez Côté, processus ascendant et processus descendant . . . 39
2.7 Représentation de l'état interne du système Perceptro après un cycle ascendant . . . 41
2.8 Instance de réseau de neurones transparent proposé par Snoussi Maddouri et coll. pour la reconnaissance de l'écriture arabe manuscrite . . . 42
2.9 Exemple de Perceptron multicouche . . . 44
2.10 Exemple de comportement en deux dimensions pour différentes valeurs de pas d'apprentissage . . . 48
2.11 Arrêt par la méthode de validation croisée . . . 49

3.1 Schéma général du réseau de neurones perceptif . . . 54
3.2 Proposition de contexte pour un article scientifique . . . 57
3.3 Schéma spécifique du RNP pour l'analyse de structures logiques d'articles scientifiques . . . 58
3.4 Classification de couleurs par un réseau de neurones perceptif . . . 60
3.5 Classification de couleurs par un Perceptron multicouche . . . 61
3.6 Cycles perceptifs et RNP, extraction, propagation, analyse, correction . . . 62
3.7 Correction de boîte englobante . . . 63
3.8 Ambiguïté sur un bloc mal segmenté . . . 64
3.9 Choix d'une hypothèse et correction de la segmentation par utilisation de l'échantillon type . . . 65
3.10 Utilisation du contexte pour lever une ambiguïté . . . 66

3.11 Document comportant une erreur d'étiquetage . . . 69
3.12 Document parfaitement labellisé . . . 69
3.13 Utilisation des normes ALTO, TEI et METS pour l'échange des données . . . 71

4.1 Séparation du XOR par l'utilisation de deux variables non informatives . . . 78
4.2 Principe général d'une méthode à adaptateur . . . 79
4.3 Approximation d'une image RVB par la SVD . . . 82
4.4 Images des vecteurs propres pour le caractère 'A' par Yanadume et coll. . . . 83
4.5 Types de connexions entre les niveaux et les neurones . . . 84
4.6 Reconnaissance du RNP avec partitionnement de l'espace d'entrée . . . 86
4.7 Partitionnement des P_i . . . 89
4.8 Vue schématique de la méthode de partitionnement . . . 90
4.9 Méthode de l'éboulis de Cattell . . . 92
4.10 Localisation des variables de sous-ensembles de différentes tailles par la méthode de partitionnement . . . 94

5.1 Réseau à quatre neurones complètement bouclé . . . 105
5.2 Vue schématique d'un réseau à décalage temporel . . . 111
5.3 Topologie du réseau de neurones dynamique perceptif . . . 113
5.4 Utilisation d'un réseau récurrent et intégration du retour de contexte dans le calcul de l'activation . . . 114
5.5 Version récurrente avec retour d'état dans les couches précédentes . . . 118
5.6 Représentation d'un ensemble de termes par une chaîne, un arbre et un graphe orienté, étiqueté et acyclique . . . 119
5.7 Architecture générale du réseau de Küchler et Goller . . . 119
5.8 Réseau d'encodage pour un graphe acyclique et pour un graphe avec cycle . . . 120
5.9 Proposition d'architecture pour la classification des structures par Sperdutti et Starita . . . 121

A.1 Page 1 du premier article . . . 130
A.2 Page 2 du premier article . . . 131
A.3 Page 5 du premier article . . . 132
A.4 Page 6 du premier article . . . 133
A.5 Les deux premières et les deux dernières pages du second article scientifique . . . 134
A.6 Les deux premières et les deux dernières pages du troisième article scientifique . . . 135
A.7 Différence entre l'image numérisée et l'image d'origine vectorielle . . . 136
A.8 Vue tridimensionnelle de la matrice de covariance de la base d'apprentissage des articles sientifiques . . . 137

B.1 Exemple d'échantillons de la base MNIST reconnus par un PMC . . . 139
B.2 Exemple d'échantillons de la base MNIST non reconnus par un PMC . . . 140
B.3 Exemple de création d'un représentant d'une classe par un algorithme génétique et utilisation d'un PMC . . . 140
B.4 Vue tridimensionnelle de la matrice de covariance de la base d'apprentissage de la base MNIST réduite à des images de 7×7 pixels . . . 142
B.5 Vue tridimensionnelle de la matrice de covariance de la base d'apprentissage de la base MNIST composées d'images de 28×28 pixels . . . 142

C.1 La sigmoïde et sa dérivée . . . 145
C.2 Exemple de descente de gradient sur une quadrique . . . 146
C.3 Exemple de sur-apprentissage . . . 147

Liste des tableaux

1.1 Niveaux de traitement selon la prédominance du contenu selon Nagy 7
1.2 Synthèse des références utilisant des systèmes à base de règles 13
1.3 Synthèse des références utilisant des règles et une représentation en arbre 16
1.4 Synthèse des références utilisant des grammaires 18
1.5 Synthèse des références utilisant un apprentissage 20
1.6 Synthèse des références utilisant des approches dirigées par les données 22
1.7 Synthèse des références utilisant des systèmes à approche ascendante ou descendante 24
1.8 Résumé d'une vue d'ensemble de plusieurs algorithmes d'analyse de la structure logique (base de données utilisée, métrique de performance, résultats obtenus et idée clé de l'algorithme) 26
1.9 Résumé détaillé de plusieurs algorithmes d'analyse de la structure logique (analyse de l'erreur, représentation des structures physique, logique et des sorties, étiquettes logiques et domaine d'application) 27

2.1 Comparaison des fonctions d'activation de McClelland et Rumelhart et de Côté et coll. 40

3.1 Indices physiques fournis par l'OCR et servant d'entrée au système de reconnaissance . . . 55
3.2 Éléments de structure logique choisis pour la base d'articles scientifiques 56
3.3 Résumé des différences majeures entre le système Perceptro et le réseau de neurones perceptif 67
3.4 Classification de structures logiques par un PMC et un RNP avec cycles perceptifs 68
3.5 Résultats détaillés du réseau de neurones perceptif au troisième cycle pour chaque classe 70

4.1 Différence entre réduction et sélection de variables 76
4.2 Interprétation des vecteurs propres pour une image 83
4.3 Résultats de reconnaissance sur la base MNIST en fonction de différentes tailles de sous-ensembles de variables 93
4.4 Résultats normalisés de reconnaissance sur la base MNIST redimensionnée en fonction de différentes tailles de sous-ensembles de variables 94
4.5 Résultats normalisés de reconnaissance de structures logiques à partir de caractéristiques physiques en fonction de différentes tailles de sous-ensembles de variables 95
4.6 Caractéristiques physiques choisies pour un premier sous-ensemble de taille dix 95
4.7 Information conservée par l'ACP en fonction de la base et du nombre de composantes principales retenu 96

4.8 Taux de reconnaissance de structures logiques en fonction de différentes tailles de sous-ensembles de variables et de méthodes de détermination de dimension de l'espace réduit 96
4.9 Classification de structures logiques par un PMC et un RNP avec cycles perceptifs et partitionnement de l'espace d'entrée 98
4.10 Résultats détaillés du réseau de neurones perceptif et partitionnement au troisième cycle pour chaque classe 99

5.1 Classification de structures logiques par un PMC, un réseau de neurones perceptif et son extension dynamique, avec cycles perceptifs et partitionnement de l'espace d'entrée 116
5.2 Résultats détaillés du réseau de neurones dynamique perceptif pour chaque classe 117

B.1 Résultats obtenus sur la base MNIST pour différents classifieurs 141

Liste des algorithmes

1 Apprentissage d'un PMC par rétropropagation du gradient 47
2 Principe de l'OBD et de l'OBS 75
3 Résumé de la méthode de partitionnement 90
4 Méthode de centres mobiles 97
5 Méthode de la puissance itérée 154
6 Méthode de la déflation 154
7 Méthode de Crout 155
8 Méthode de Rutishauser 156
9 Itération k de la méthode QR 156
10 Itération k de la méthode QR avec translation 156

Introduction

Cette thèse porte sur l'élaboration d'une approche générique pour l'analyse de structures logiques d'images de documents imprimés à partir d'informations extraites de la structure physique. Basé sur des mécanismes cognitifs, le système proposé, appelé réseau de neurones dynamique perceptif, adopte une reconnaissance flexible et évolutive par le biais de l'utilisation de cycles perceptifs qui permettent de traiter des formes ambiguës en effectuant des retours sur les entrées et en corrigeant les données fautives ou en rajoutant d'autres données le cas échéant.

L'utilité d'un document tient autant de l'information que l'on extrait de sa lecture que de la possibilité de réutiliser cette information. L'utilisation des traitements de texte a permis d'uniformiser la présentation des documents et de faire ressortir les entités logiques essentielles du texte dans un format réutilisable. Disposer à la fois du texte brut et des informations logiques l'accompagnant permet d'effectuer des traitements plus complets et plus élaborés comme la recherche sélective et structurée [Clarke et coll., 1995 ; Piwowarski et coll., 2002], l'indexation, la création de résumés [Alam et coll., 2003], l'interprétation [Ceheux, 2002], la réédition, la reformulation et le reformatage vers d'autres supports électroniques [Quint et Vatton, 1986 ; Belaïd et coll., 2004 ; Belaïd et coll., 2005].

La production d'un tel document n'est pas chose aisée. Quand le document est produit localement et que ses sources sont encore disponibles, il est généralement facile d'extraire et de réutiliser l'information, bien que souvent la tâche nécessite des traitements avancés et spécifiques. Dans les autres cas, l'extraction demeure plus délicate. Le document est soit écrit dans un format spécifique comme le PDF imposant une conversion dans le format local désiré [Bloechle et coll., 2006], soit imprimé. Pour cette dernière situation, mise à part une saisie manuelle fastidieuse, la solution commune consiste à numériser et enregistrer le document sous la forme d'une image et à le traiter par un système d'analyse et de reconnaissance d'images de documents.

La mise en place d'un tel système nécessite plusieurs étapes et celle de la reconstruction de la structure logique n'intervient que tardivement dans la chaîne complète d'analyse [Ingold, 2002]. Le postulat que nous nous posons comme la plupart des autres auteurs est de considérer les étapes de prétraitement, d'analyse de la structure physique et de la reconnaissance du texte comme déjà effectuées.

Les contributions dans la littérature utilisent essentiellement des méthodes basées sur le modèle. Les systèmes à base de règles, d'arbres de décision, de systèmes experts ou de grammaires formelles occupent une place importante dans la reconstruction de la structure logique. L'inconvénient de ces systèmes vient du fait qu'un modèle de document doit être fourni et que les règles qui découlent du modèle deviennent vite arbitraires et ne peuvent pas couvrir un

vaste ensemble de documents. Les règles sont en même temps très restrictives vis-à-vis de la classe à laquelle elles sont destinées ; dès que les entrées du système sont bruitées, incomplètes ou contradictoires elles échouent très vite. Les méthodes à base de grammaires formelles ont, à l'inverse, l'avantage de limiter les types de production pouvant être générés et par conséquent restreignent aussi les règles que le langage peut satisfaire. Elles sont plus souples et donnent de meilleurs résultats mais il n'en reste pas moins qu'elles restent très dépendantes du modèle de classe considéré et nécessitent souvent une analyse physique de bonne qualité et à un niveau très fin allant jusqu'à l'extraction complète du texte.

A contrario, les approches dirigées par les données, comme les réseaux de neurones, sont extrêmement peu représentées dans la reconnaissance de structures logiques, elles sont pourtant capables d'absorber les différentes perturbations des données d'entrée et de découvrir, par un apprentissage, les liaisons entre les observations physiques et leur interprétation logique. En dépit des travaux effectués sur des tâches préliminaires comme le prétraitement, la reconnaissance de la structure physique et du texte, les contributions utilisant des approches neuronales n'ont pas retenu autant d'attention dans la phase nous intéressant.

En tout état de cause, l'utilisation directe de réseaux comme le Perceptron multicouche pour une analyse de formes aussi structurées que peuvent l'être les éléments logiques ne semble pas être une alternative viable. De plus, plusieurs auteurs concluent qu'une intégration de connaissances semble indispensable à la création d'un système performant. Une approche totalement dirigée par les données ne serait donc pas une solution générique. L'idée développée dans cette thèse est de déterminer comment utiliser une approche neuronale qui conserverait les propriétés d'une approche dirigée par les données et qui permettrait une intégration de connaissances sur le document et qui adapterait sa reconnaissance en fonction de la difficulté de la tâche à accomplir.

Il est apparu que les travaux de [McClelland et Rumelhart, 1981] sur un modèle d'activation interactive soient une partie de la réponse au problème d'hybridation soulevé. Bien que conçus pour un cadre applicatif différent, celui de la reconnaissance de l'écriture, les principes perceptifs mis en avant sont transposables à notre domaine.

Dans [Côté, 1997 ; Côté et coll., 1998], les auteurs proposent un système de reconnaissance de l'écriture cursive nommé Perceptro utilisant des concepts perceptifs proposés par [McClelland et Rumelhart, 1981]. Il est fondé sur un modèle connexionniste avec traitement parallèle de l'information, un mécanisme interactif d'activation et une décomposition de la connaissance en trois niveaux d'interprétation. Il sera repris et modifié avec succès par [Snoussi Maddouri et coll., 2002] sous le nom de réseau de neurones transparent.

La reconnaissance s'effectue en des séries de deux phases successives, les cycles perceptifs : l'une ascendante permettant de faire remonter les informations de bas niveau vers le niveau supérieur (primitives physiques vers les mots), l'autre descendante qui fait descendre les informations de haut niveau vers le niveau inférieur (mots vers primitives) avec un retour de contexte permettant de corriger la segmentation initiale des lettres dans l'image.

Les concepts mis en place dans cette approche sont séduisants du fait que les mécanismes d'interaction entre les niveaux d'interprétation permettent une exploitation plus aisée des connaissances contextuelles à différents niveaux. De plus, la nature hiérarchique de la structure logique, la nécessité d'intégrer de la connaissance pour aider l'interprétation et surtout la correction de la segmentation sont autant de points communs qui nous confortent dans l'idée d'utiliser une topologie et une reconnaissance par cycles similaires.

Le modèle que nous proposons se nomme réseau de neurones dynamique perceptif. Il reprend les principes fondateurs des précédents auteurs à savoir l'intégration de concepts à chaque neurone, une organisation en couches d'interprétation et une reconnaissance par cycles perceptifs. Les changements se feront au niveau de la fonction d'activation et de la détermination par apprentissage des poids du réseau. Un parallèle de l'utilisation du contexte (l'effet de la supériorité du mot sur la lettre) sera fait avec l'organisation hiérarchique de la structure logique.

L'idée des cycles perceptifs sera conservée, nous utiliserons aussi plusieurs couples de propagation ascendante et descendante pour décider de la classe d'une forme. La segmentation en blocs du document sera elle aussi corrigée, tout comme celle des lettres chez les précédents auteurs, avec toutefois une technique différente pour aider le système à calibrer les nouvelles segmentations.

Un autre apport sera fait au niveau de la sélection des variables d'entrée. Afin de limiter le nombre d'extractions à effectuer à chaque cycle, nous proposons une méthode de partitionnement de l'espace d'entrée ayant pour but la création de sous-ensembles de variables qui seront donnés progressivement au réseau. Elle reposera sur une sélection par filtre, allant au-delà d'un simple classement de variables, par l'étude des interactions des groupes de variables entre eux. Ceci nous procurera à la fois des sous-ensembles informatifs et contenant le moins de redondance possible à l'intérieur de chacun.

Afin de bénéficier de la puissance des cycles perceptifs, nous proposons aussi une nouvelle architecture du réseau de neurones qui intègre la composante temporelle pendant l'apprentissage et la reconnaissance. Sur la base d'un réseau à décalage temporel, nous nous sommes appropriés le concept de ligne de temporisation pour exploiter les informations provenant de précédents cycles dans l'évaluation du cycle courant. La version dynamique permet de mieux tenir compte de la variabilité des données lorsqu'elles sont corrigées après chaque cycle, les réponses du réseau sont plus adéquates en fonction de l'avancement de la reconnaissance.

Le réseau de neurones dynamique perceptif a été testé sur une base d'articles scientifiques, les résultats obtenus sont supérieurs à ceux d'une approche neuronale classique et sont comparables voire meilleurs que ceux utilisant des méthodes uniquement dirigées par le modèle. Bien que les outils d'extraction d'observations physiques soient modestes, le potentiel de la méthode est effectif et peut être aussi largement amélioré en s'intéressant à de meilleures variables d'entrée et surtout en renforçant l'aspect dynamique et récurrent de la topologie du réseau.

La thèse est organisée de la façon suivante :

- le premier chapitre présente un état de l'art des méthodes d'analyse et de reconnaissance d'images de documents en opposant les approches dirigées par le modèle de celles dirigées par les données. Nous montrerons l'intérêt d'utiliser une approche hybride pour bénéficier des qualités de chacune ;
- le deuxième chapitre se focalisera sur l'utilisation des méthodes neuronales dans l'analyse de documents. Bien qu'elles soient presque exclusivement employées pour les tâches précédant celle de la reconnaissance de structures logiques, nous montrerons comment peuvent être résolues ces tâches, que nous considérons comme déjà effectuées, et nous apporterons un aperçu de l'efficacité de ces méthodes. De plus en se focalisant sur des travaux menés en psychologie cognitive, nous montrerons comment certains systèmes neuronaux à représentation locale peuvent être utilisés pour résoudre notre problème avec l'aide de cycles perceptifs et de l'information de contexte. Nous finirons par une description du Perceptron multicouche qui fera le lien entre le modèle à représentation locale des précédents auteurs et le système dirigé par les données que nous proposons ;

- le troisième chapitre présente le réseau de neurones perceptif. Nous montrerons comment il a été adapté au cas de la reconnaissance de structures logiques par la modification de l'architecture et l'intégration du contexte. Nous exposerons aussi les apports comme l'apprentissage, l'utilisation d'échantillons type pour la correction de données d'entrée ;
- le quatrième chapitre apporte une amélioration du temps de reconnaissance par l'utilisation d'un partitionnement de l'espace d'entrée. Le réseau sera alimenté par des sous-ensembles de variables et le nombre de variables présentées sera en relation avec la complexité de la forme à reconnaître ;
- le dernier chapitre exposera une amélioration apportée au réseau de neurones perceptif qui consiste à l'étendre à une version dynamique, avec l'intégration de la composante temporelle. Sur la base d'un réseau à décalage temporel, nous avons exploité la ligne de temporisation pour intégrer, dans l'apprentissage et la reconnaissance, les informations des précédents cycles perceptifs afin d'adapter la décision au cycle courant ;
- le thèse se termine par une conclusion et des perspectives qui montreront les évolutions possibles de ce système.

Chapitre 1

Analyse et reconnaissance d'images de documents

Dans ce chapitre, nous présentons un état de l'art des méthodes d'analyse et de reconnaissance d'images de documents. Nous nous intéresserons dans un premier temps à définir les notions que nous utiliserons dans la suite du manuscrit ainsi que le cadre d'étude. Nous étudierons plusieurs méthodologies de reconnaissance en distinguant principalement les méthodes dirigées par le modèle de celles dirigées par les données. La critique qui sera faite en fin de chapitre exposera les difficultés à résoudre et montrera aussi comment la méthode que nous proposons se positionne dans la littérature.

Sommaire

1.1 Définitions 5
1.2 Reconnaissance de documents 6
1.3 Structure physique, structure logique 8
1.4 Méthodologies pour la reconnaissance de documents 11
1.4.1 Approches dirigées par le modèle 11
1.4.2 Approches dirigées par les données 20
1.4.3 Approches descendantes et ascendantes 22
1.5 Évaluation des performances 24
1.6 Discussion des méthodes 25
1.7 Conclusion 28

1.1 Définitions

Depuis l'apparition des premiers systèmes de lecture optique de caractères il y a plus de quarante ans, le thème de l'analyse d'images de documents[1] n'a cessé de se développer. Ce thème de recherche fait partie du domaine du traitement d'images numériques qui avait pour objectif principal de convertir des images de documents en vue de la modification, l'archivage, la recherche, la réutilisation et la transmission de l'information que ces images contiennent.

[1] *« L'analyse d'images de documents est une théorie et une pratique de reconstruction de la structure symbolique des images numériques directement produites par l'ordinateur ou numérisées à partir du papier »* [Nagy, 2000]

Suivant les auteurs, on retrouvera plusieurs désignations se référant à des phases précises lors d'un processus complet d'analyse. Il est assez difficile de définir de manière très formelle les notions utilisées dans la littérature car les frontières sont floues et se chevauchent. Plusieurs termes sont employés et peuvent être classés dans un ordre partant du traitement du plus bas niveau proche de l'image vers le plus haut niveau logique comme suit :

- l'analyse physique de documents concerne l'aspect visuel du document (la mise en page). Elle se compose classiquement d'une première étape de prétraitement suivie d'une étape d'extraction de caractéristiques physiques décrivant les blocs de l'image sans pour autant chercher encore à leur attribuer une signification ;
- la reconnaissance de documents qui consiste essentiellement à extraire le texte de l'image et les structures logiques (comme les titres, auteurs, sommaire, etc.) l'accompagnant, afin de retrouver à la fois le contenu et le sens de chaque bloc ;
- l'interprétation de documents détermine les relations entre les composants logiques en rattachant par exemple la légende à la figure associée ou bien encore la détermination de l'ordre de lecture ;
- la compréhension de documents faisant référence à l'analyse plus sémantique du contenu du document. Elle va au-delà de la simple identification de texte en effectuant des traitements plus avancés comme la construction de résumés automatiques.

Nous nous intéressons à la reconnaissance de documents et plus précisément à la détermination d'éléments de la structure logique. Dans l'approche que nous proposons, le problème est posé en terme de reconnaissance des formes ; nous partons des résultats de l'analyse physique de l'image pour retrouver les étiquettes de chaque objet c'est-à-dire le nom de son correspondant dans la structure logique. Nous nous situons à une étape charnière d'un processus global de reconnaissance qui permet de faire le lien entre la description purement géométrique de l'image et une interprétation plus approfondie du contenu textuel.

1.2 Reconnaissance de documents

Sous le terme générique de reconnaissance d'images de documents, trois grandes applications peuvent être distinguées :

- le document manuscrit [Vinciarelli, 2002 ; Nicolas et coll., 2004], où l'écriture n'est pas conventionnelle et où la reconnaissance se limitera à la détection de lignes et de mots qui restent des tâches extrêmement difficiles ;
- les documents à prédominance graphique comprenant plusieurs thématiques comme par exemple l'interprétation de symboles [Lladós et coll., 2001b ; Delalandre et coll., 2003], l'analyse de tableaux et de formulaires [Cesarini et coll., 2003], [Amano et coll., 2004], l'analyse de cartes et de plans [Burge et Monagan, 1995] ou bien encore l'analyse de documents techniques [Wenyin et Dori, 1999] ;
- les documents éditoriaux structurés [Summers, 1995b ; Nagy, 2000], qui ont une structure physique régulière, composée de blocs rectangulaires et disposant d'une structure logique assez standardisée. L'analyse de ces documents est souvent appelée rétro-conversion qui se veut le processus inverse de celui de l'édition.

Nous nous restreindrons à la famille des documents structurés, notre partie expérimentale sera orientée vers des documents de type articles scientifiques qui sont générés principalement en LaTeX et suivant des contraintes d'édition imposées par les différents éditeurs. En fonction du type de document à analyser, les traitements classiques à effectuer diffèrent mais peuvent être mis

en parallèle car les problèmes rencontrés, tout comme les techniques permettant leur résolution sont en général similaires. Dans [Nagy, 2000], l'auteur propose une division des traitements en cinq niveaux de granularité (Tab. 1.1).

Niveau du traitement	*Document structuré*	*Document graphique*
Pixels	**Prétraitement**	**Prétraitement**
	Représentation	Représentation
	Réduction de bruit	Réduction de bruit
	Binarisation	Binarisation
	Détection de l'inclinaison	Squelettisation
	Segmentation de zones informatives	Vectorisation
	Segmentation de caractères	
	Reconnaissance de la fonte et de la langue	
Primitives	**Reconnaissance de glyphe**	**Reconnaissance de primitives**
	Composantes connexes	Segments de droites et de courbes
	Lignes, ponctuation et diacritiques	Jonctions et nœuds
	Mots	Caractères
Structures	**Reconnaissance de texte**	**Reconnaissance de structure**
	Segmentation en mots	Champs texte
	Reconstruction des lignes de texte	Légendes
	Analyse de tableau	Attribution d'étiquette
	Contexte morphologique	Dimensions
	Contexte lexical	Symboles graphiques
	Syntaxe, sémantique	Caractéristiques de surface et texture
Documents	**Analyse de la structure de page**	**Interprétation**
	Séparation texte/graphique	Reconnaissance d'objet
	Analyse des composantes physiques	Analyse de la connexité
	Analyse des composantes logiques	Séparation en couches CAO/SIG
	Composantes fonctionnelles (étiquetage)	Extraction attributs base de données
	Compression	Compression
Corpus	**Recherche d'information**	**Base de données, CAO, SIG**
	Classification de documents et indexation	Validation
	Recherche	Recherche
	Sécurité, authentification	Mise à jour

TABLEAU 1.1 – Niveaux de traitement selon la prédominance du contenu [Nagy, 2000]

Nous nous plaçons à un niveau macrostructurel de l'analyse en ne détaillant pas certains problèmes spécifiques et particuliers au niveau de la microstructure. Il existe en effet certaines formes qu'il est possible de rencontrer dans l'image du document et qui nécessitent à elles seules une analyse particulière pour être décomposées plus finement. Nous ne couvrirons donc pas par exemple l'interprétation des :

- formes mathématiques [Kacem et coll., 1999 ; Kosmala et coll., 1999 ; Zanibbi et coll., 2002] [Garain et coll., 2004a ; Garain et coll., 2004b ; Toyozumi et coll., 2004] ;
- tableaux [Turolla et coll., 1995 ; Hu et coll., 2002 ; Ramel et coll., 2003 ; Zhang et coll., 2007] et des formulaires [Watanabe et coll., 1995 ; Belaïd et coll., 1998] ;

– logos [Francesconi et coll., 1997 ; Kim et Kim, 1998 ; Chang et Chen, 2001 ; Doermann et coll., 1996] ;
– bibliographies [Parmentier et Belaïd, 1997 ; Besagni et coll., 2003 ; Belaïd et coll., 2000] ;
– ou bien encore des tables de matières [Belaïd et Toussaint, 2000 ; Tsuruoka et coll., 2001].

Le cadre d'étude est focalisé sur le problème d'association d'étiquettes logiques à chaque élément de la structure physique. Nous nous situons donc au centre d'un processus complet d'analyse de documents ; nous partons des résultats de l'analyse physique qui seront nos entrées, nos sorties seront les entrées des méthodes d'interprétation. Il est à noter que l'étape d'étiquetage logique ne constitue pas à elle seule le processus de reconnaissance car sous cette dénomination, et suivant les auteurs, on peut retrouver l'indexation [Doermann, 1998], la recherche de régions d'intérêt [Casasent et coll., 1997] ou bien encore la structuration et la correction des données [Weindorf, 2001].

1.3 Structure physique, structure logique

Nous avons déjà évoqué dans la section précédente les notions de structure physique et de structure logique sans les définir. Bien que la notion de structure de documents soit une notion assez subjective [Ingold, 1989], on retrouvera très souvent la distinction entre physique et logique. La différence entre les deux réside dans le critère choisi pour diviser le document : la structure physique est fondée sur la présentation visuelle du contenu (on parle aussi de structure géométrique) tandis que la structure logique est basée sur l'organisation du texte et dépend de la compréhension par l'humain du contenu [Brown, 1989]. Concernant les documents éditoriaux, elles décrivent une organisation hiérarchique et récursive du contenu du document en des parties de plus en plus fines, elles sont typiquement représentées sous forme d'arbres, reflétant la hiérarchie des objets les composant (Fig.1.1) [André et coll., 1989].

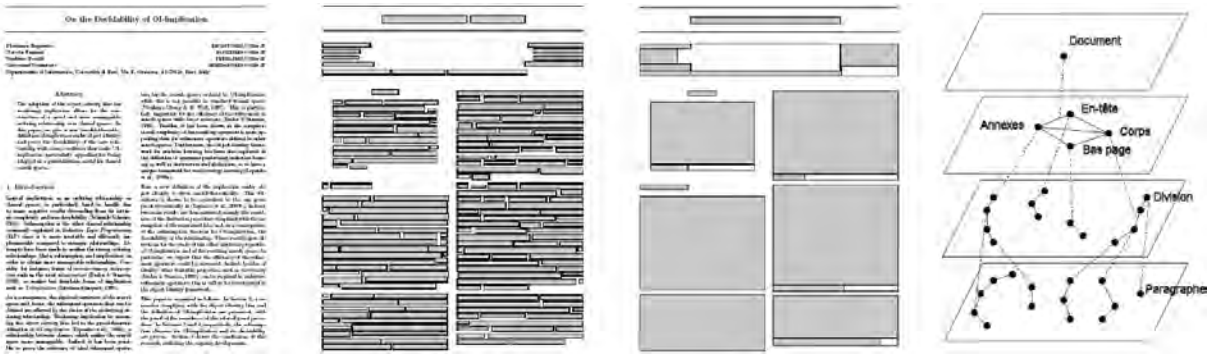


FIGURE 1.1 – Exemple de document à gauche, suivi de deux décompositions physiques à deux niveaux de granularité (mots et blocs) et représentation en arbre d'une décomposition logique du document

La représentation de ces structures est aussi un problème majeur car chaque système tente d'imposer son propre formalisme de représentation ce qui les rend incompatibles entre eux. Des normes ont été proposées pour harmoniser les différents échanges de fichiers entre les différents systèmes comme DAFS [Dori et coll., 1995] mais n'ont pas été retenues par la communauté.

La norme ODA[2] propose un modèle hiérarchique orienté objet pour représenter les documents et sépare les structures physique et logique (Fig. 1.2). Dans le même ordre d'idée, la norme SGML propose aussi un modèle hiérarchique pour décrire les structures mais contrairement à la norme ODA, il s'agit d'un langage à balisage employant une syntaxe rigoureuse pour décrire les structures et le contenu des composantes logiques d'un document défini dans une DTD *(Document Type Definition)*.

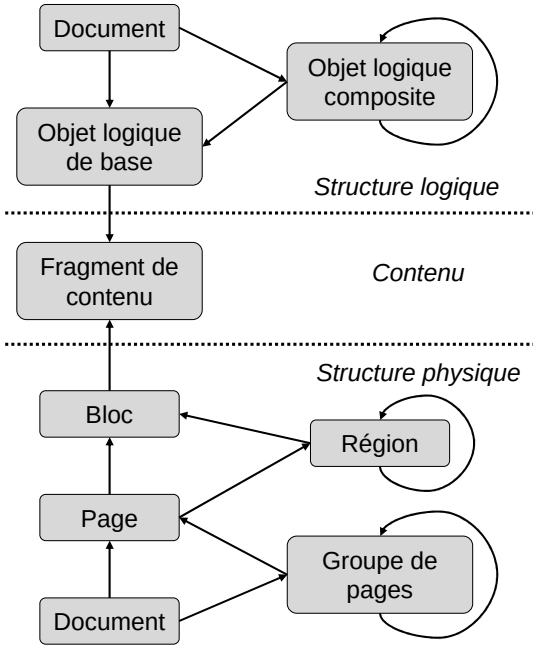


FIGURE 1.2 – Représentation hiérarchique des structures de documents [Belaïd, 1997]

Plus récemment, deux normes proposent de représenter indépendamment les deux structures : ALTO[3] pour la structure physique et TEI[4] pour la structure logique. Les deux sont des schémas XML, organisant une fois de plus de manière hiérarchique le document. Le modèle ALTO a été conçu pour représenter les informations provenant de la reconnaissance par OCR de pages de documents comme les livres, les revues ou les journaux. Toute information, géométrique ou de style, apportée par un OCR trouve une balise dans le langage ALTO, son approche descendante commence au niveau de la page et découpe les éléments jusqu'au niveau de la lettre où il encode même le score de confiance de l'OCR donné pour la lettre. La TEI est une DTD SGML accompagnée d'un volume de recommandations *(TEI guidelines)* expliquant de quelle façon doit être utilisée la DTD pour représenter récursivement le contenu d'un document. Bien que principalement adaptée aux besoins de la communauté des chercheurs en sciences humaines, elle s'est modularisée en proposant un grand nombre de DTD pour différentes classes de documents comme par exemple les articles scientifiques, les dictionnaires ou bien encore les proses. Comme

[2]ISO8613, Information Processing, Text and Office Systems, Office Document Architecture (ODA) and Interchange Format, Parts 1,2,4-8, 1989

[3]Analyzed Layout and Text Object, The Library of Congress, National Digital Newspaper Project, `http://www.ccs-gmbh.com/alto/`

[4]Text Encoding Initiative, TEI Consortium Guidelines, `http://www.tei-c.org/`

les deux structures sont dissociées et stockées dans des fichiers séparés, le standard METS[5] se propose de conserver les liens entre les deux structures et également avec l'image. Il se compose de pointeurs écrits en XML qui mettent en relation un objet d'une structure avec un ou plusieurs objets de l'autre structure. L'association de ces trois standards que nous avons retenue est capable de décrire entièrement les documents que nous manipulons.

```
<TEI.2>
    <teiHeader> [ header information for the composite ] </teiHeader>
    <text>
        <front> [ front matter for the composite  ]      </front>
        <group>
           <text>
              <front> [ front matter of first text ] </front>
              <body>  [ body of first text  ]          </body>
              <back>  [ back matter of first text ]    </back>
           </text>
           <text>
              <front> [ front matter of second text]  </front>
              <body>  [ body of second text  ]          </body>
              <back>  [ back matter of second text ]    </back>
           </text>
           [ more texts or groups of texts here ]
        </group>
        <back>      [ back matter for the composite  ]      </back>
    </text>
</TEI.2>
```

FIGURE 1.3 – Composition générale d'un fichier TEI

L'analyse de la structure physique fait intervenir plusieurs étapes incluant le prétraitement de l'image, la décomposition de l'image de chaque page en blocs puis la classification de ces blocs en fonction de leur contenu (texte, graphique ou image) en vue de reconstruire leur organisation. Des articles de synthèse comme [Nadler, 1984 ; Srihari et Zack, 1986 ; Belaïd et coll., 1993 ; Cattoni et coll., 1998] donnent un aperçu des techniques employées et des difficultés à résoudre. Nous considérons cette partie du travail d'analyse comme déjà effectuée, nous nous limiterons à une analyse primaire donnée par les OCR en sachant que des méthodes bien plus adaptées existent (S.-Sec. 2.4.4, p.49). Nous ne posons pas le postulat que les données physiques soient parfaitement extraites, nous tiendrons compte des imperfections ou de l'absence des informations provenant de l'analyse physique.

L'analyse de la structure logique est une tâche encore plus difficile car elle dépend davantage de l'application et de la classe de documents visées. Un élément de structure logique peut avoir des sens différents en fonction de l'interprétation que se fait l'utilisateur du document (Fig. 1.4). Elle dépend aussi techniquement de l'analyse de la structure physique qui est la condition préalable nécessaire à la majorité des systèmes de reconnaissance [Haralick, 1994]. La reconnaissance de structures logiques s'appuie sur les informations physiques trouvées au préalable et il s'agit d'étiqueter les blocs conformément à la structure physique et au modèle.

Les travaux menés en reconnaissance de documents se focalisent généralement uniquement sur la phase d'étiquetage, le problème d'extraction des relations entre les composants logiques comme l'ordre de lecture ou encore les références croisées sont beaucoup moins traités [Malerba

[5] Metadata Encoding and Transmission Standard, The Library of Congress, `http://www.loc.gov/standards/mets/`

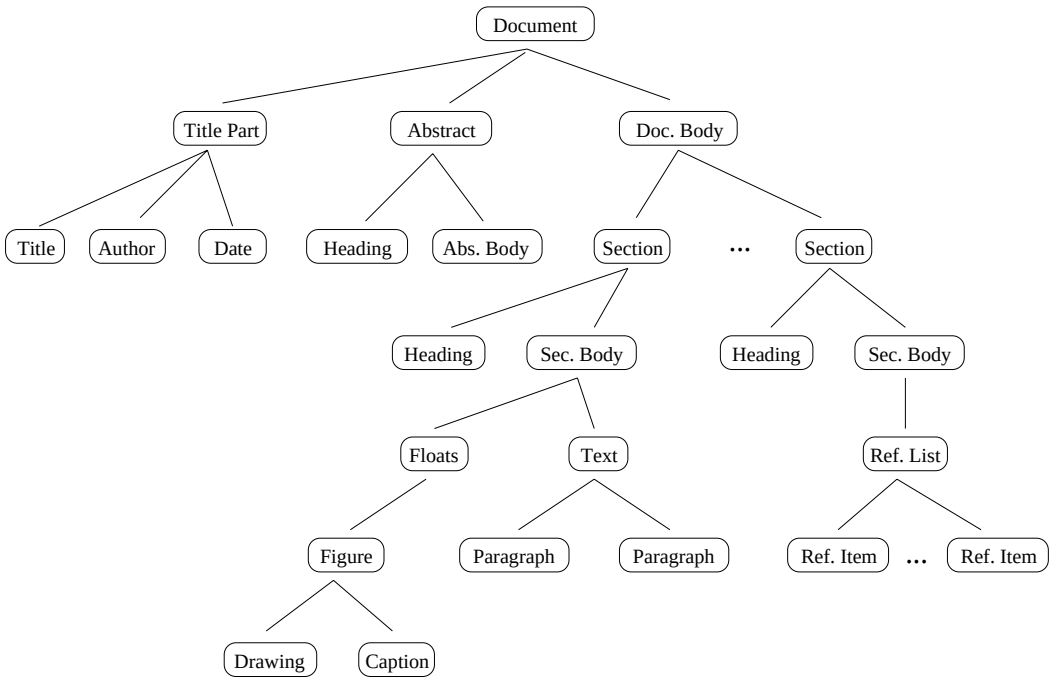


FIGURE 1.4 – Instance de structure logique pour un article scientifique selon [Summers, 1995b]

et coll., 2007] du fait de la nécessité d'interprétation du contenu qui est une tâche différente et utilisant d'autres approches dédiées à cette finalité.

1.4 Méthodologies pour la reconnaissance de documents

Deux familles de méthodes sont employées pour l'analyse de la structure logique [Tang et coll., 1999]. L'approche dirigée par le modèle *(model-driven method)*, prédominante dans la littérature, qui s'oppose à l'approche dirigée par les données *(data-driven method)*, beaucoup moins présente. Une classification similaire consiste à séparer l'approche statistique ou connexionniste de celle syntaxique ou structurelle. Dans les approches dirigées par les données, on retrouvera par exemple des méthodes connexionnistes faisant appel à des données statistiques de l'image, les sorties du réseau donnant les étiquettes des structures logiques reconnues. Pour les approches dirigées par le modèle, la littérature propose un grand nombre de solutions utilisant des outils syntaxiques comme les grammaires pour représenter le modèle (connaissances a priori) et des analyseurs syntaxiques pour inférer la reconnaissance.

1.4.1 Approches dirigées par le modèle

Dans ce type d'approche, un modèle de document est nécessaire au système, lui procurant les hypothèses de reconnaissance. Si l'on s'intéresse aux formalismes prédominants dans les approches dirigées par le modèle tels que les analyseurs syntaxiques ou les systèmes experts, il

est nécessaire de disposer d'un ensemble de règles qui spécifient les liens entre les observations physiques et les étiquettes à assigner aux blocs. Dans le cas de l'utilisation d'un ensemble de règles, il n'y a pas de description des relations sémantiques entre les composantes logiques. L'utilisation des arbres (également issus d'un ensemble de règles) rend la description des relations plus aisée car la structure d'arbre traduit de manière plus directe la structure du modèle. Son parcours fait appel plus généralement à des méthodes de parcours de graphes [Hancock et Wilson, 2002].

Systèmes à base de règles

Les systèmes à base de règles comme montrés dans [Lin et coll., 1997 ; Ishitani, 1999 ; Kim et coll., 2001] utilisent classiquement des suites de conditions pour identifier chaque élément de structure logique mais ignorent les relations entre les éléments logiques. Les performances restent très bonnes lorsque les entrées physiques sont conformes aux règles prédéfinies. [Lin et coll., 1997] atteignent environ 97% de reconnaissance pour six éléments de structure (texte, titre, image, numéro de page, en-tête et légende) dans 235 pages de livre, les règles sont du type : *Si un bloc est le plus en bas de la page et si son contenu est un nombre, alors le bloc est le numéro de page.* [Ishitani, 1999] obtient 96,3% sur 150 documents variés avec 10 structures à reconnaître (titre principal, en-têtes, notes de bas de page, légendes, notes, programmes, titres, paragraphes, listes et formules), [Kim et coll., 2001] obtiennent 96% pour l'extraction de quatre zones d'intérêt (titre, auteurs, affiliations, résumé) à partir de 120 règles (Fig. 1.5) sur une base de 11 000 articles médicaux.

RULE 1
1. Number_Headtitle==0
2. Font_Size==Max_Font_Size
3. Number_Degree<3 or Percent_Degree<10
4. Number_Middlename<3 or %_Middlename<10
5. Coordinate_Upper<Height_Article/3
6. Coordinate_Lower<Height_Article/2
7. If all of above conditions are satisfied {
If (Font_Size==Max_Font_Size) PID=100
Else If (|Font_Size - Max_Font_Size|<3) PID=99
Else PID=(Font_Size -
Min_Font_Size)'100/(Max_Font_Size -
Min_Font_Size)
} Else {PID=0}

RULE 2
If (PID<100) pick a zone having the highest PID for title.
RULE 3
1. Distance from a zone to title is smaller than that of any other labels.
2. Font_Size, Font_Attribute, Med_Line_Height, and Med_Line_Space of a zone must be similar to those of title zone.
RULE 4
Coord_Upper of title<Coord_Upper of author<Coord_Upper of affiliation<Coord_Upper of abstract

FIGURE 1.5 – Exemple de règles pour la détection du titre [Kim et coll., 2001]

Dans [Kreich et coll., 1991], les auteurs se servent de deux bases de connaissance : l'une pour la structure logique et l'autre pour la structure physique. Pour chaque forme à reconnaître, une mise en correspondance (par une métrique de Hamming) est effectuée et produit un score de confiance. La détermination d'une étiquette logique se fait principalement par l'extraction de mots-clés à l'intérieur du texte reconnu. [Summers, 1995a] utilise un principe voisin ; il compare les données obtenues par l'extraction de la structure physique à des prototypes prédéfinis. En fonction d'une mesure de ressemblance avec les prototypes, une étiquette peut être affectée à chaque forme. Dans les expérimentations, 16 classes sont utilisées, chacune dispose de plusieurs prototypes

pour mieux les représenter. Les résultats reportés sont de l'ordre de 86% sur des documents scientifiques, ce qui est très correct compte tenu du nombre de classes et de la proximité de certaines. Des scripts sont employés par [Dengel et Klein, 2002] pour reconnaître des régions d'intérêt dans des images de factures médicales. Pour des formes complexes, des expressions régulières sont couplées à des thésaurus pour retrouver l'étiquette à partir du texte contenu dans la zone d'intérêt. Un solveur de contraintes est aussi utilisé en tant que coordinateur final pour vérifier la fiabilité du résultat de chaque script. En fonction du type de document traité, les résultats s'échelonnent de 73% pour des documents hors classe jusqu'à 100% pour des factures de dentistes. Le taux moyen est de 92% pour six structures et 525 documents.

Auteurs	Méthodes	Résultats
[Lin et coll., 1997]	Règles	97%
[Ishitani, 1999]	Règles	96,3%
[Kim et coll., 2001]	Règles	96%
[Kreich et coll., 1991]	Règles et métrique	non communiqué
[Summers, 1995a]	Règles et prototypes	86%
[Dengel et Klein, 2002]	Scripts et solveur	92%

TABLEAU 1.2 – Synthèse des références utilisant des systèmes à base de règles

Systèmes à base de règles et représentation arborescente

Afin de faire intervenir les relations qui existent entre les objets logiques, certains auteurs introduisent une représentation en arbre de la structure physique et logique. Le système DeLoS [Niyogi et Srihari, 1995] utilise ce principe pour dériver une structure arborescente de la structure logique. Il repose sur des bases de connaissance des deux structures ainsi que sur un système de règles qui se charge de la création de l'arbre final. Les règles ne sont pas seulement des transcriptions d'observations physiques vers l'interprétation logique ; elles incluent aussi d'autres niveaux d'analyse comprenant des règles de stratégie qui guident le système afin d'utiliser les règles à bon escient, ainsi que des règles de contrôle pour évaluer le résultat de l'application des autres règles de transformation (Fig. 1.6). Pour 13 pages d'un journal quotidien, les auteurs obtiennent 82% de bonne classification avec 160 règles pour un nombre non annoncé de structures logiques. Dans [Fisher, 1991] plusieurs familles de règles (positionnement, formatage et textuelles) sont aussi distinguées mais aucun résultat expérimental n'est donné.

La représentation en arbre est aussi utilisée par [Tsujimoto et Asada, 1990], un ensemble de règles réalise la transformation de l'arbre physique en arbre logique. Un taux d'environ 93% pour sept structures (titre, résumé, sous-titres, paragraphe, en-tête, numéro de page, légende) est obtenu sur une centaine de documents courants (articles, magazines, journaux, etc.). Dans [Yamashita et coll., 1991], l'approche est assez similaire : elle a pour objectif la gestion efficace des cas de contradiction. Pour ce faire, une méthode de relaxation est mise en œuvre pour éliminer les étiquettes incorrectes. Les trois quarts des 77 documents utilisés (des brevets) sont correctement étiquetés. Des transformations d'arbres similaires sont aussi réalisées en parallèle de l'extraction de l'arbre physique par [Derrien-Péden, 1991] en se basant principalement sur la décroissance de l'espacement entre des blocs pour déduire la profondeur d'un bloc dans la hiérarchie logique.

```
IF a block Z is of type "large-text",
  OR IF it satisfies the following three conditions:
      {it is of type "medium-text",
         AND it is below another block W,
         AND block W is not of type "large-text"
               or "medium-text"},
THEN block Z is a major headline.
```

(a) Knowledge Rule

```
IF the grouping mode is on,
  AND a block has been selected,
THEN find all the immediate neighbors
       of the selected block.
```

(b) Control Rule

```
IF any partially grouped units remain,
THEN apply all unit-related control rules
       for each of these units
     until there are no more partial units.
```

(c) Strategy Rule

FIGURE 1.6 – Exemple de règles utilisées par [Niyogi et Srihari, 1995]

Une méthode de logique floue est utilisée par [Hu et Ingold, 1993] et permet de résoudre certains problèmes liés aux erreurs et incertitudes des OCR. Elle permet aussi de trouver des similarités entre une entrée et un modèle candidat. L'approche est descendante et consiste à parcourir le graphe du document et à étudier tous les chemins possibles pour étiqueter les nœuds. Plusieurs stratégies sont discutées et les auteurs proposent l'emploi d'une analyse mixte de programmation dynamique qui selon eux permet de résoudre les cas difficiles.

Un grand nombre de systèmes réalise dans un premier temps une étape d'analyse de la structure physique. On retrouve assez fréquemment le *XY-cut* [Nagy et Seth, 1984] pour construire une première hiérarchie physique. C'est le cas par exemple de [Ishitani, 2003] qui l'utilise pour un découpage récursif de l'image. Dès que cette première structure est trouvée, le principe de la solution proposée consiste à s'approcher d'un modèle pivot prédéfini représentant la structure logique. L'ensemble des données est représenté en XML puis un parseur utilise des transformations de type scripts XSLT pour reconstruire la structure logique. Il obtient un taux de 95,2% sur 150 documents de bureau avec dix structures mais uniquement sur la labellisation, il n'y pas de résultat sur la qualité de la structure logique extraite.

Le système ΠODA de [Dengel et coll., 1992] utilise une approche assez similaire où le savoir est codé dans un arbre appelé *geometric tree*. Il s'apparente à un arbre de décision où les nœuds encodent une règle géométrique (basée sur l'information physique) et les feuilles les étiquettes logiques. En parcourant l'arbre avec les données physiques et ce jusqu'à la feuille, on retrouve une étiquette pour chaque bloc. Ici aussi les règles sont données par un expert (190 échantillons ont dû être observés pour écrire les règles permettant de reconnaître 11 structures logiques).

Dans [Wenzel et Maus, 2001], le formalisme de représentation de la connaissance est différent : au lieu de stocker des heuristiques générales pour tout type de document, les auteurs ont fait le choix de répartir les connaissances dans des cadres *(frames)* qui sont des parties spécifiques de documents. En outre, les connaissances génériques d'un document sont placées dans des *frames* spéciales appelées concepts. L'étiquetage se fait ensuite par un analyseur de règles qui prend en compte toutes les *frames* ainsi que les relations entre elles. Les résultats en terme de précision sont de l'ordre de 90% pour 3 structures.

[Saitoh et coll., 1993] proposent un système pour la segmentation, la classification de blocs et l'ordre de lecture. La méthode est ascendante et détermine en parallèle de la segmentation physique les étiquettes logiques des régions d'intérêt. L'ordre de lecture est déduit d'une part de l'arbre construit durant la segmentation grâce aux relations père-fils et de règles basées sur la notion «d'intervalle d'influence» des blocs entre eux.

Une approche mixte entre ascendante et descendante est utilisée par [Chenevoy et Belaïd, 1991] où le système d'analyse utilise une technique de tableau noir appelée Graphein. L'idée est d'identifier la structure spécifique d'un document à partir d'un modèle générique (Fig. 1.7). Le système peut traiter différentes hypothèses de structuration et peut prendre en compte le contexte structurel des documents. Il dispose d'une base de connaissances décrivant la structure physique et logique du document. Le tableau noir sert à faire coopérer des processus spécialisés dans la segmentation et la reconnaissance de formes par un mécanisme de gestion d'hypothèses. Plusieurs constructeurs sont définis pour créer la structure logique et des qualifieurs renseignent sur les occurrences d'un objet ou sur ses caractères optionnel, obligatoire, répétitif, etc. tels que définis dans la norme ODA.

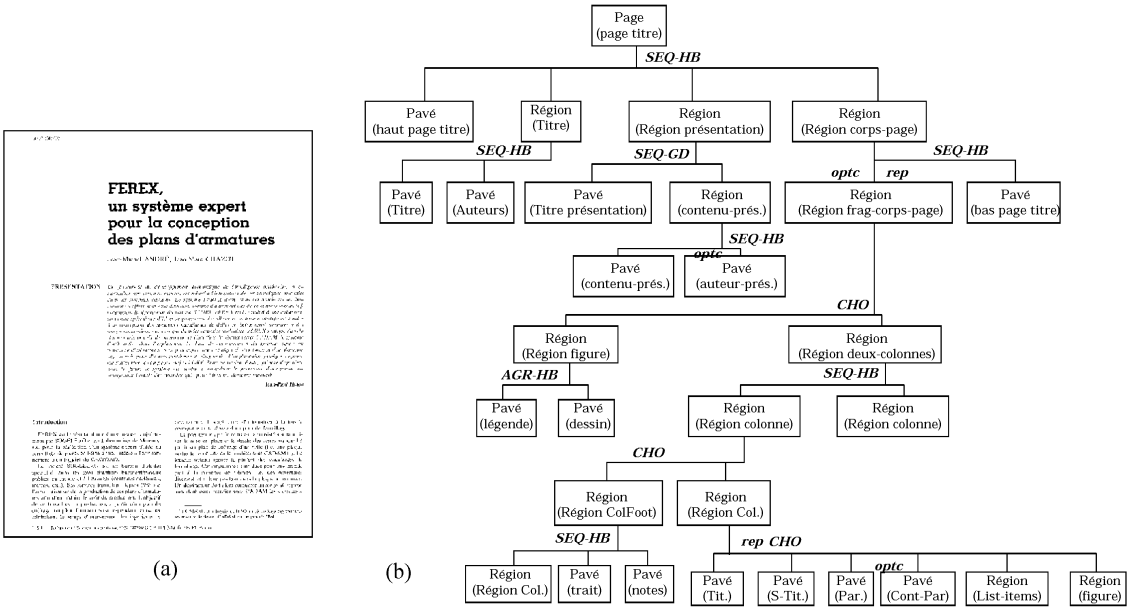


FIGURE 1.7 – Exemple de modélisation dans Graphein d'une page d'un article de revue par un modèle physico-logique générique

Auteurs	Méthodes	Résultats
[Niyogi et Srihari, 1995]	arbres, règles	82%
[Fisher, 1991]	arbres, règles	n.c.
[Tsujimoto et Asada, 1990]	arbres, transformation d'arbres	93%
[Yamashita et coll., 1991]	arbres, relaxation	75%
[Derrien-Péden, 1991]	transformation d'arbres	n.c.
[Hu et Ingold, 1993]	arbre, logique floue	n.c.
[Ishitani, 2003]	*XY-cut*	95,2%
[Dengel et coll., 1992]	arbre de décision	n.c.
[Wenzel et Maus, 2001]	connaissances spécialisées	90%
[Saitoh et coll., 1993]	règles d'influence	n.c.
[Chenevoy et Belaïd, 1991]	tableau noir	n.c.

TABLEAU 1.3 – Synthèse des références utilisant des règles et une représentation en arbre

Systèmes à base de grammaires

Les méthodes souvent utilisées et donnant de bons résultats sont celles à base de grammaires formelles dans lesquelles la structure du document est considéré comme une phrase composée soit d'étiquettes logiques soit d'une suite d'observations de caractéristiques physiques du document.

Un algorithme de segmentation hiérarchique de pages est décrit dans la publication de [Krishnamoorthy et coll., 1993]. Les auteurs construisent un arbre dans lequel chaque nœud représente un bloc de l'image. Ils utilisent un arbre X–Y étiqueté pour stocker à la fois la structure physique et la structure logique. Des grammaires sont ensuite spécifiées pour chaque bloc ce qui permettra d'étiqueter l'arbre logique. Il est à noter qu'en présence de bruit, comme pour toutes les approches similaires, leur algorithme d'analyse syntaxique peut échouer sur des données erronées ou incomplètes. De plus, comme aucune fonction objectif n'est minimisée, l'analyse n'est donc généralement pas optimale. Une quarantaine de grammaires sont nécessaires pour obtenir un étiquetage complet de 9 documents sur 12, les trois autres étant étiquetés à 69%, 73% et 93%.

Des grammaires de type BNF étendues sont utilisées par [Ingold et Armangil, 1991]. Les règles y sont séparées en deux catégories : les règles de composition pour définir la structure logique générique et les règles de présentation qui établissent les caractéristiques physiques des structures logiques à reconnaître. Le document est entièrement représenté par un graphe d'analyse dans lequel le système cherche un chemin pour étiqueter la structure logique en respectant les contraintes des attributs physiques.

Dans [Conway, 1993], l'auteur décrit la structure physique par une grammaire bidimensionnelle permettant de conserver les relations topologiques (haut, bas, gauche et droite) dans les voisinages (Fig. 1.8). Elle est similaire à une grammaire hors-contexte et un parseur est utilisé pour analyser syntaxiquement les pages segmentées selon la grammaire. Seuls des résultats de temps d'exécution absolus sont donnés.

Les approches à base de grammaires standard sont généralement déterministes ce qui leur vaut un comportement peu fiable sur des documents bruités. Dans la pratique, elles s'appliquent difficilement à des documents réels et il est d'autant plus difficile de lever certaines ambiguïtés que l'extraction des composantes physiques est de mauvaise qualité.

TitlePage → (**over** Title Author Organisation Body)
Body → (**leftside** Column Column)
Column → (**over** ParaBody **?** (Paragraph | Figure)* Footnote **?**)
Paragraph→(**over** textline <first_indented> ParaBody)
Parabody→(**over** textline<aligned>*)
Title→(**over** textline[large,bold]<centred>+)
over$(x, y) \Leftrightarrow o(x, y) \wedge \neg(\exists z[o(x, z) \wedge o(z, y)]$

FIGURE 1.8 – Exemple de grammaire utilisée par [Conway, 1993]

Il existe cependant des adaptations permettant de tenir compte d'entrées bruitées tout en conservant la performance d'une grammaire. Une grammaire stochastique est utilisée par [Tateisi et Itoh, 1994] qui consiste à tenir compte d'un ensemble de coûts pondérant chaque règle grammaticale. Les coûts déterminent la probabilité qu'une règle soit appliquée dans un cas d'ambiguïté. L'analyse syntaxique est alors stochastique et le système retient les résultats possibles ordonnés par leur probabilité. Le nombre de structures à reconnaître n'est pas donné mais les auteurs obtiennent jusqu'à 89% de bon étiquetage sur des documents scientifiques.

Dans [Coüasnon, 2006], une méthode de description de document et de modification de segmentation est proposée pour reconnaître des formes structurées. Le système nommé DMOS est constitué d'un langage grammatical singulier *(Enhanced Position Formalism)* permettant de décrire une structure générique tout en l'associant à un analyseur syntaxique spécifique, capable de travailler sur des données bruitées afin de reconnaître, par compilation, les structures logiques. L'idée clé consiste à définir une méthode générique qui peut s'instancier pour divers types de documents (partition musicale, formule mathématique, formulaires) ; c'est un système qui génère un autre système spécifique s'adaptant à la tâche de reconnaissance en changeant uniquement les connaissances a priori. Avec le formalisme EPF, il est possible de générer des descriptions spécifiques très précises du document même si l'image d'entrée est fortement dégradée. L'EPF se détache d'une grammaire traditionnelle par la possibilité d'introduire du contexte dans la segmentation du document et ne se contente pas ainsi uniquement de vérifier syntaxiquement si une segmentation valide ou non un modèle. La robustesse de l'approche est testée sur des formulaires militaires anciens du 19^{e} siècle très dégradés, les résultats de reconnaissance sont de 98,82% pour une base de près de 88 000 images et avec un taux de rejet de 1,18%.

De manière similaire à ce qui est fait pour les systèmes à base de règles, il est fréquent d'appliquer des grammaires sur des structures de graphes [Blostein et coll., 1996] plutôt que sur des chaînes. Au lieu de faire un appariement entre graphes extraits et graphes modèles comme dans les méthodes de parcours, il est possible d'utiliser l'ensemble des règles de transformation (la grammaire) afin de corriger et faire converger les graphes candidats vers les graphes modèles. Le problème se transforme alors en un problème de comparaison de sous-graphes inexacts [Lladós et coll., 2001a]. Cette approche est par exemple utilisée dans [Nagy et coll., 1992] où les auteurs utilisent une analyse syntaxique pour déterminer l'étiquetage logique. En émettant l'hypothèse que les conventions de mise en page soient connues et fixes, des règles sont extraites pour retrouver les étiquettes en fonction des informations trouvées durant l'analyse physique. Ces informations sont ensuite codées en chaîne de symboles pour permettre une analyse syntaxique. Les résultats sont qualitatifs mais les auteurs estiment que lors du test, 9 pages sur 12 ont été parfaitement labellisées.

Auteurs	Méthodes	Résultats
[Krishnamoorthy et coll., 1993]	grammaire, arbre	95%
[Ingold et Armangil, 1991]	grammaire, graphe	n.c.
[Conway, 1993]	grammaire bidimensionnelle	n.c.
[Tateisi et Itoh, 1994]	grammaire pondérée	89%
[Coüasnon, 2006]	grammaire EPF	98,82%
[Nagy et coll., 1992]	comparaison de graphes et grammaire	75% de pages parfaites

TABLEAU 1.4 – Synthèse des références utilisant des grammaires

Systèmes à apprentissage

L'utilisation de connaissances est très présente dans les systèmes dirigés par le modèle mais, dans les précédentes citations, cette connaissance est directement intégrée dans les règles et doit être donnée par un expert. Pour s'affranchir de cette étape, [Brugger et coll., 1997 ; Brugger et coll., 1998] ont proposé un système dont le modèle peut être appris à partir d'exemples. Ils utilisent une N-grammaire (avec $N = 3$) pour décrire la structure logique du document. La reconnaissance se fait en construisant un arbre logique au-dessus des données physiques tout en respectant les contraintes du modèle de classe. La conformité de l'arbre est évaluée par une mesure heuristique qui permet de valider des séquences d'observations physiques en fonction de l'arbre logique, du nombre de formes dans l'arbre et d'une probabilité d'apparition de trigramme. La construction complète de l'arbre se ramène à un problème d'optimisation que les auteurs résolvent par un algorithme de type *best-first*. L'apprentissage du modèle se fait de manière incrémentale : après avoir intégré quelques arbres déjà générés, le système est capable d'effectuer une reconnaissance sur un nouveau document. Si l'analyse échoue, même partiellement, l'utilisateur corrige de manière interactive l'arbre qui est ensuite renvoyé au système d'apprentissage. Après une mise à jour des probabilités, il est intégré au modèle. Un prototype a été expérimenté sur des pages d'agenda.

Une N-grammaire et une représentation arborescente ont aussi été utilisées par [Hurst, 2001]. L'auteur se concentre sur un problème plus réduit qui consiste à séparer les tables des autres objets dans les images de documents. Les résultats sont de l'ordre d'un peu moins de 99% pour la précision[6] et 80% pour le rappel[7].

Une solution alternative, pouvant s'apparenter à un prétraitement d'une méthode *model-driven*, est d'extraire le modèle à partir d'exemples comme [Akindele et Belaïd, 1995] le proposent. Ils estiment qu'un modèle de classe générique de document est une connaissance importante pour l'analyse de documents. Ils proposent d'apprendre un modèle générique à partir d'exemples appartenant à une même classe. La méthode est basée sur une inférence de grammaire d'arbre et construit un modèle à travers des constructeurs de type ODA. La méthode utilise dans un premier temps un modèle initial, faisant intervenir l'utilisateur, qui contrôle et valide les échantillons qui seront utilisés pour inférer le modèle générique. Un premier test de validité est effectué pour séparer les structures reconnues et non reconnues en comparant l'arbre du modèle générique et l'arbre spécifique du document en cours de traitement. Si la structure spécifique est acceptée, elle est étiquetée en suivant le modèle générique, dans l'autre cas, l'utilisateur doit

[6] rapport du nombre de formes pertinentes trouvées au nombre total de formes trouvées
[7] rapport du nombre de formes pertinentes trouvées au nombre total de formes pertinentes

assigner une étiquette au nœud de l'arbre posant problème. L'inférence du modèle générique utilise la méthode de [Gonzalez et coll., 1976] qui produit une grammaire d'arbre à partir d'un ensemble d'échantillons d'arbres qui se compose d'une étape d'extraction des règles d'expansion, d'une fusion de règles équivalentes et d'une réduction éliminant la redondance et la répétition. Les résultats du système sont un ensemble de règles décrivant le modèle générique de la classe de document correspondant aux échantillons utilisés. Les règles générées correspondent à celles construites manuellement.

Un apprentissage incrémental est mis en avant dans [Hadjar et coll., 2002]. La contribution décrit une méthode de reconnaissance de documents à structure complexe permettant une construction incrémentale du modèle dans un environnement interactif. Les auteurs utilisent les interactions de l'utilisateur pour améliorer leur système en construisant une base de connaissances : chaque correction de la segmentation change la caractéristique discriminante pour la forme corrigée. Après 150 opérations manuelles, le système obtient des taux d'environ 85% pour six classes à séparer sur 29 pages de journaux.

Une méthode à base de graphes est proposée par [Liang et Doermann, 2002]. La représentation se fait en fonction de l'extraction des blocs physiques : un nœud du graphe caractérise un bloc (position, taille, etc.), les arcs reflètent les relations spatiales entre les blocs (en dessous, à gauche, etc.). Pour chaque classe de document, un modèle de graphe est déterminé portant des informations supplémentaires avec l'étiquette logique et d'autres attributs qui pondèrent les arcs du graphe. Une fonction de coût adaptée à la structure est utilisée pour évaluer la proximité de deux graphes. Pour déterminer la correspondance entre un graphe extrait d'une image et le graphe modèle, les auteurs utilisent un algorithme de type *branch and bound* pour trouver rapidement la solution. L'apprentissage des modèles est plus à proprement parler une post-correction : un graphe est d'abord établi manuellement, si un nouveau document n'est pas reconnu par le modèle, les poids de ce dernier sont mis à jour pour prendre en compte les différences entre lui et le nouveau document. Le taux d'erreur moyen est inférieur à 10% pour 160 pages de titres d'articles scientifiques et pour une dizaine d'étiquettes. Les résultats sont obtenus après dix cycles d'apprentissage. Une approche similaire est reportée par [Héroux et coll., 2000].

Pour les systèmes à base de règles, [Esposito et coll., 2004] proposent d'utiliser différentes méthodes inductives pour retrouver des règles à partir d'un ensemble de documents d'entraînement provenant d'archives. La démarche d'apprentissage n'est pas entièrement automatique et la reconnaissance reste quand même dépendante d'un expert fixant certaines règles que ce soit au niveau de l'analyse logique et encore plus pour la compréhension de documents. Le système se nomme Wisdom++ et a déjà été utilisé dans d'autres applications comme les documents scientifiques [Altamura et coll., 2001].

Le système DAVOS de [Dengel et Dubiel, 1996] dispose lui aussi d'un module d'apprentissage. Ce sont des concepts qui sont extraits automatiquement en repérant des valeurs d'attributs singulières dans les objets composant le document. L'apprentissage non supervisé a pour objectif de caractériser les similarités et les différences pour chaque élément de structure. L'ensemble des concepts est représenté dans un arbre géométrique *(GTree)*. Cette hiérarchie de concepts est ensuite prise comme référence pour classifier les futures entrées. Les tests sont effectués sur une centaine d'images d'enveloppes, pour dix classes à reconnaître, les meilleures obtiennent une précision de 95% (corps, logo et pied) et la plus mauvaise 38% (expéditeur).

Le terme d'apprentissage incrémental utilisé en analyse de documents doit être pris parfois avec précaution car son utilisation diffère de celle que l'on peut trouver par exemple dans le vocabulaire des réseaux de neurones. Dans la majorité des travaux, le système initial n'est pas entraîné, l'opération d'ajout de nouvelles informations n'est pas toujours automatique. La

convergence des systèmes est peu discutée et les critères d'arrêt théorique ne sont pas donnés. On pourra qualifier l'apprentissage de construction incrémentale ou interactive [Liu et coll., 2002].

Auteurs	Méthodes	Résultats
[Brugger et coll., 1997]	n-grammaire, optimisation	n.c.
[Hurst, 2001]	n-grammaire, arbres	99% de précision
[Akindele et Belaïd, 1995]	inférence de grammaire d'arbre	n.c.
[Hadjar et coll., 2002]	correction de segmentation	85%
[Liang et Doermann, 2002]	graphe, optimisation	90%
[Héroux et coll., 2000]	graphe, optimisation	n.c.
[Esposito et coll., 2004]	induction, règles	n.c.
[Altamura et coll., 2001]	induction, règles	n.c.
[Dengel et Dubiel, 1996]	concepts, arbre	95%
[Liu et coll., 2002]	règles, apprentissage guidé	96,5%

TABLEAU 1.5 – Synthèse des références utilisant un apprentissage

1.4.2 Approches dirigées par les données

En opposition avec les approches dirigées par le modèle décrites dans la sous-section précédente, on considère ici que le système doit découvrir seul les règles de passage entre la structure physique et la structure logique. Il doit classiquement partir de l'information physique brute pour effectuer la reconnaissance ou se doit d'extraire seul la connaissance nécessaire à la conversion par une autre méthode. Les approches basées sur les données sont principalement des systèmes de type fouille de données ou bien encore des systèmes utilisant l'apprentissage automatique.

S'il est fréquent de retrouver dans des méthodes dirigées par le modèle la présence de modules qui pourraient faire partie d'approches dirigées par les données, il est assez rare de trouver un système complètement dirigé par les données et exempt de toute information a priori.

Il existe des systèmes ayant une prédominance neuronale comme celui de [Sainz Palmero et coll., 1996 ; Sainz Palmero et Dimitriadis, 1999], qui ont développé un algorithme d'apprentissage neuronal flou *(RFasArt : Recurrent Fuzzy Adaptive System ART)*. Il classe, pour chaque nouveau bloc physique, les étiquettes logiques candidates et choisit la meilleure parmi les solutions admissibles. La base de test comprend 102 lettres (91% de reconnaissance sur 7 structures) et 14 papiers scientifiques (97% sur 9 structures). [Aiello et coll., 2002] proposent également une approche neuronale qui utilise un système d'apprentissage C4.5 [Quinlan, 1993] d'arbre de décision pour apprendre les règles de classification qui servent ensuite à reconnaître les composantes textuelles. Les tests sont effectués sur près de 800 pages (majoritairement des articles scientifiques). Sur les quatre structures logiques à reconnaître, la précision est en moyenne de 95% pour les articles et 85% pour les hétérogènes.

[Le Bourgeois et coll., 2001 ; Souafi-Bensafi et coll., 2002] proposent une comparaison entre une relaxation probabiliste [Rosenfeld et coll., 1976], des réseaux bayésiens [Pearl, 1988] et des champs de Markov afin de reconnaître les structures logiques. L'idée est de considérer la reconnaissance de n'importe quel bloc en fonction des autres blocs trouvés dans son voisinage. Le système est appliqué dans les expérimentations à la reconnaissance de tables des matières et

il atteint au maximum 95% en utilisant la relaxation sur 10 pages. Les auteurs projettent une combinaison des trois méthodes pour améliorer les résultats.

D'autres travaux essaient de reconsidérer la nature structurelle de la tâche à accomplir : [Walischewski, 1997] propose de représenter chaque structure de document comme un graphe attribué direct (chaque sommet étant un élément de structure) qui caractérise la fréquence d'apparition de différentes relations spatiales. Le graphe est ensuite appris par un algorithme incrémental qui met à jour les nœuds et les étiquettes après chaque passage d'échantillon. Les expérimentations sont menées sur 1000 lettres administratives et un taux moyen de 95% de reconnaissance est obtenu pour quatre classes (expéditeur, destinataire, date et corps de document).

Dans [Ceci et coll., 2005], les auteurs partent du constat que la majorité des méthodes proposées ne considère pas l'aspect multirelationnel existant lors de la reconnaissance. Deux systèmes sont évalués dans leurs travaux : Mr-SBC (classifieur bayésien structurel multirelationnel) et ATRE (programmation logique inductive multirelationnelle). Les tests sont effectués sur 22 articles scientifiques avec 16 structures logiques à découvrir. Le meilleur taux est atteint pour le numéro de page à savoir 96% mais en moyenne le taux est de 50% pour ATRE et 40% pour Mr-SBC (le nombre de faux positifs étant assez faible surtout pour ATRE).

Quand un réseau de neurones est utilisé, les auteurs poursuivent rarement l'analyse jusqu'à l'étape de reconstruction de la structure logique. Par exemple, [Le et coll., 1995a ; Le et coll., 1995b] utilisent plusieurs réseaux neuronaux (Perceptron multicouche (PMC), réseau à fonction de base radiale, réseau probabiliste et carte auto-organisatrice) pour traiter l'analyse physique de l'image. Les auteurs se concentrent sur l'orientation de la page, la correction de l'inclinaison et la classification des blocs en texte et non-texte. Pour cette dernière phase, c'est le réseau à fonction de base radiale qui obtient les meilleurs résultats à savoir 99,6% sur une base de test d'un peu moins de 800 blocs. Les autres réseaux sont très proches de ce score (le PMC obtient 94,4%) mais aucune perspective n'est donnée pour éventuellement augmenter le nombre de classes à séparer.

Les réseaux de neurones artificiels comme le Perceptron multicouche sont extrêmement rares en tant que méthode d'analyse de la structure logique, certains auteurs les utilisent comme des outils de combinaison de méthodes classiques. C'est le cas par exemple de [Azzabou et Likforman-Sulem, 2004] qui utilisent un PMC pour combiner les résultats d'un premier système effectuant une analyse physique de l'image et d'un second système à base de règles effectuant une analyse textuelle du contenu. L'apprentissage permet de remplacer une fonction de score pondérant les résultats des outils d'extraction avec une assignation manuelle des coefficients. L'ensemble du système est testé sur une reconnaissance de lettres administratives et il obtient des résultats assez fiables (25% de précision et 71% de rappel).

Dans le domaine de l'extraction de cellules de formulaires, [Belaïd et coll., 1998] proposent une méthodologie pour combiner les résultats de plusieurs PMC à partir d'indices physiques, comme l'orientation du texte et de la morphologie des caractères, pour étiqueter des cellules de formulaires parmi huit classes. Une première classification est effectuée pour séparer les formes en fonction d'un ensemble de caractéristiques numériques (comme le nombre de composantes connexes, l'alignement du texte, le nombre de lignes, la densité de pixels noirs, etc.). Une deuxième classification est réalisée par deux PMC suivant les résultats obtenus dans l'étape précédente en se basant cette fois-ci sur l'image normalisée des cellules afin de séparer chaque forme. La reconnaissance est de 91% pour un ensemble de 3 500 cellules issues de 19 formulaires.

Auteurs	Méthodes	Résultats
[Sainz Palmero et Dimitriadis, 1999]	apprentissage neuronal flou	97%
[Aiello et coll., 2002]	arbre de décision	95%
[Le Bourgeois et coll., 2001]	relaxation probabiliste	95%
[Walischewski, 1997]	graphe, apprentissage incrémental	95%
[Ceci et coll., 2005]	classifieur bayésien, logique inductive	50%
[Le et coll., 1995a ; Le et coll., 1995b]	réseaux de neurones	99,6%
[Azzabou et Likforman-Sulem, 2004]	Perceptron multicouche	25%
[Belaïd et coll., 1998]	PMC	91%

TABLEAU 1.6 – Synthèse des références utilisant des approches dirigées par les données

1.4.3 Approches descendantes et ascendantes

Dans les méthodes d'analyse et reconnaissance de documents, deux grandes familles d'approches peuvent qualifier le fonctionnement de l'algorithme en s'intéressant à quel niveau le système débute sa reconnaissance. On parlera d'approches descendantes si l'on décompose de plus en plus finement l'analyse et d'approches ascendantes si la démarche inverse est entreprise à savoir partir d'une information de bas niveau en agglomérant de proche en proche les éléments similaires pour reconstruire des zones homogènes de plus grande taille.

Dans le cas d'une méthode descendante, on essaye d'approcher la structure en se fondant sur un modèle dont on extrait les caractéristiques qui sont nécessaires pour être en mesure de le valider. C'est le cas de la technique utilisée par [Wolf et coll., 1997]. Dans leur application, ils divisent des images d'enveloppes en blocs pour lesquels ils calculent un indice d'homogénéité. Après l'extraction de plusieurs caractéristiques sur chaque bloc ou agrégation de blocs, un test de plausibilité est effectué pour déterminer si la zone correspond à l'adresse du destinataire. Dans [Tang et coll., 1997], l'image en niveau de gris est subdivisée en plusieurs morceaux par une décomposition en ondelettes. À partir de cette décomposition, les lignes de références sont facilement extraites et servent à la construction de la structure physique. Pour classifier des formes structurées, [Esposito et coll., 1992] utilisent aussi une approche descendante basée sur la logique des prédicats. L'idée clé est de calculer une mesure d'adéquation flexible entre une description symbolique et les observations bruitées récupérées. Les bases d'apprentissage et de test sont constituées respectivement de 40 et 35 documents de différents types parmi lesquels 30 sont bien classifiés.

Les approches ascendantes extraient d'abord un certain nombre de caractéristiques sur l'image d'entrée pour ensuite tenter de les mettre en correspondance avec le modèle. Ce principe est utilisé par [Gyohten et coll., 1995] qui extraient des caractères japonais dans des documents non formatés et sans connaissance a priori sur la structure physique. La méthode utilise un schéma multiagent dans lequel la coordination des agents permet d'obtenir une extraction uniquement à partir de données locales sur les composantes connexes. Une approche similaire est utilisée dans [Parmentier et Belaïd, 1997] qui utilisent un réseau de concepts pour représenter la structure générique et où plusieurs agents sont mis à contribution pour faire émerger la structure logique.

Il n'existe pas réellement de méthode suffisamment générique pour couvrir un large ensemble de types de documents. Il semblerait que les approches ascendantes soient en général plus efficaces mais sous réserve de pouvoir estimer de manière fiable la structure physique à l'avance. Elles sont en contrepartie souvent plus coûteuses à cause des extractions parfois inutiles sur l'image mais peuvent produire un résultat généralement robuste, homogène et ceci même dans le cas où le modèle de la structure n'est pas connu à l'avance. Elles sont aussi préférables lorsque l'agencement du document est très variable et éloigné du modèle de classe. Les approches descendantes sont davantage adaptées à des documents dont on peut prédire la structure. Elles privilégient la vue globale du document et ne subdivisent les zones qu'en cas de nécessité.

Dans des documents présentant de nombreux points fixes (zones facilement repérables par leur position) comme les cartes de visite, les enveloppes ou la table des matières, il est possible de guider les outils d'analyse en fonction du type de la zone à traiter et éventuellement de générer des hypothèses pour aider la découverte par exemple des lignes ou des lettres s'il s'agit d'un OCR [Belaïd et coll., 2000].

La classification entre approche descendante et ascendante n'est pas tout à fait pertinente dans le cas de l'analyse du logique. Elle est héritée de l'analyse de la structure physique et plus particulièrement de la segmentation de pages [O'Gorman et Kasturi, 1996] pour laquelle il est plus approprié de classifier les algorithmes dans les deux familles. Dans ce cadre, l'approche descendante faisant référence aux méthodes découpant itérativement l'image en blocs de plus en plus petits [Wahl et coll., 1982 ; Fletcher et Kasturi, 1988 ; O'Gorman, 1993 ; Jain et Yu, 1998 ; Kise et coll., 1998] et l'approche ascendante commençant au niveau du pixel, les agrègent en composantes connexes puis en blocs de plus grande taille, jusqu'à la zone voulue [Baird et coll., 1990].

En fait, un grand nombre de méthodes d'analyse de la structure logique se font en parallèle de l'extraction de la structure physique et principalement en utilisant une méthode de type *XY-cut* de [Nagy et coll., 1992]. Il existe aussi des méthodes hybrides utilisant les deux approches comme le proposent [Wang et Srihari, 1989 ; Pavlidis et Zhou, 1992 ; Okamoto et Takahashi, 1993 ; Etemad et coll., 1994] pour la segmentation de page.

Certains travaux ne s'intéressent pas à l'ensemble de l'image : ils ne concentrent l'analyse que sur des régions d'intérêt *(ROI)*, les autres, bien que contenant du texte et de l'information, ne sont pas analysées. Il est en effet très courant de ne chercher par exemple que le code postal dans une lettre [Cohen et coll., 1994 ; Koch et coll., 2005] ou seulement le montant d'une facture [Nielson et Barrett, 2003]. [Srihari et coll., 1999 ; Mulgaonkar, 1986] se proposent par exemple d'interpréter automatiquement l'information contenue dans les champs d'une adresse postale. Ils se servent de l'entropie de Shannon pour caractériser à la fois l'ensemble des informations fournies par chaque composant de l'adresse mais aussi les interactions entre ces composants. La stratégie de la reconnaissance consiste à utiliser l'information déjà disponible et les redondances découvertes pour trouver la valeur des composants incertains. Ils peuvent alors par exemple confirmer ou trouver certains chiffres du code postal en se basant sur le nom de la ville et de l'état. En fonction de l'objectif à atteindre, les ROI sont plus ou moins nombreuses et différentes à classifier. Dans notre application, nous n'avons pas de ROI a priori ; l'ensemble de la page doit être analysé et aucune région ne doit rester sans étiquette logique.

Auteurs	Méthodes	Résultats
[Wolf et coll., 1997]	décomposition, indice d'homogénéité	n.c.
[Tang et coll., 1997]	décomposition en ondelettes	n.c.
[Esposito et coll., 1992]	logique des prédicats	85%
[Gyohten et coll., 1995]	système multiagent	n.c.
[Belaïd et coll., 2000]	étiquetage de partie de discours	89%
[Wahl et coll., 1982] [Fletcher et Kasturi, 1988] [O'Gorman, 1993] [Jain et Yu, 1998] [Kise et coll., 1998]	descendante	segmentation de pages
[Baird et coll., 1990] [Nagy et coll., 1992]	ascendante	segmentation de pages
[Wang et Srihari, 1989] [Pavlidis et Zhou, 1992] [Okamoto et Takahashi, 1993] [Etemad et coll., 1994]	mixte ascendante et descendante	segmentation de pages
[Chenevoy et Belaïd, 1991]	tableau noir	n.c.
[Cohen et coll., 1994] [Nielson et Barrett, 2003] [Srihari et coll., 1999]	analyse syntaxique patrons entropie	région d'intérêt

TABLEAU 1.7 – Synthèse des références utilisant des systèmes à approche ascendante ou descendante

1.5 Évaluation des performances

L'évaluation des performances d'un système de reconnaissance de structures logiques est une tâche extrêmement difficile car il faut considérer plusieurs paramètres : la définition d'une métrique pour qualifier la performance, des critères de comparaison entre algorithmes, une description de la base de test (et d'apprentissage si nécessaire), la définition d'un document de vérité, l'analyse des performances mais aussi de l'erreur et du rejet.

L'ensemble des conditions à remplir pour évaluer quantitativement et qualitativement les systèmes proposés dans la littérature n'est jamais entièrement présent et équivalent d'un système à l'autre. Chaque auteur propose une métrique et l'adapte en fonction des aspects de l'algorithme qu'il veut étudier ou des conclusions qu'il veut souligner. Deux métriques sont toutefois assez fréquentes à savoir l'analyse en termes de pourcentage d'étiquetage et en termes de précision et de rappel. Dans le cas d'une métrique commune, les travaux ne portent généralement jamais sur la même base de données, ni sur le même nombre de documents ou de classes à reconnaître.

Il est difficile dans le cas de l'analyse de la structure logique de proposer une synthèse concise et significative comme il est possible de le faire pour d'autres domaines (Tab. B.1, p. 141). Certains auteurs préfèrent même une description qualitative plutôt qu'une présentation de résultats chiffrés reposant sur des notions non formalisées. Nous avons emprunté un tableau récapitulatif [Mao et coll., 2003] qui résume les expérimentations et les performances obtenues par plusieurs algorithmes d'analyse de la structure logique.

Le tableau 1.8 donne à la fois la description de la base documentaire utilisée, les performances obtenues et la métrique utilisée ainsi que l'idée principale de la méthode. Il est à noter qu'aucune des références ne donne les spécifications d'une vérité terrain et que toutes ne donnent pas de résultats chiffrés. Le tableau 1.9 présente plus en détail les représentations utilisées pour les structures physique et logique ainsi que le format de fichier final, on retrouve aussi plus précisément les étiquettes logiques à reconnaître ainsi que le domaine général d'application.

1.6 Discussion des méthodes

La majorité des méthodes proposées et citées dans la littérature sont dirigées par le modèle. On remarque qu'au final peu de contributions les utilisant reposent sur des modèles formels de document. De ce point de vue, les méthodes proposées ne sont pas assez génériques pour être appliquées à n'importe quel type de document. [Mao et coll., 2003] estiment par exemple qu'un modèle formel apporterait plusieurs avantages comme :

- choisir le niveau de complexité approprié pour un modèle étant donné la classe de document ;
- s'aider des exemples d'une classe pour estimer les paramètres du modèle ;
- utiliser le modèle pour valider des documents réels ou l'utiliser pour générer des pages synthétiques pouvant être utilisées, par la suite, dans d'autres expérimentations.

De manière générale, la création du modèle, la spécification d'une grammaire ou la détermination d'un ensemble de règles sont formellement mal définies et sont en plus données de manière empirique.

Pour certains systèmes à base de grammaire, les données d'entrée sont directement des chaînes syntaxiques représentant parfaitement la structure physique, ils ne prennent pas en compte le fait que l'extraction des indices physiques peut être imparfaite Les résultats évoqués dans ce type d'expérimentation sont alors relativement loin de la vérité terrain. Les approches dirigées par le modèle utilisent majoritairement des modes de fonctionnent déterministes. Dans le cas de données réelles pouvant contenir du bruit, des contradictions ou tout simplement de l'information manquante, les systèmes peuvent échouer et être incapables de donner une information pertinente sur la cause de l'erreur. Des mesures qualifiant la gravité de l'erreur sont rarement fournies tout comme un score de confiance sur les sorties correctes du système. La détection des ambiguïtés et le traitement du rejet se trouvent très difficiles à effectuer. La maîtrise des résultats d'un système est toujours intéressante car ce dernier peut être facilement la base d'applications semi-automatiques performantes.

Les systèmes utilisant des représentations en arbre apportent une aide lors de la reconnaissance, mais le problème d'un mauvais branchement est toujours possible et peut avoir des conséquences graves. Ils posent aussi des problèmes de construction et de maintenance. Il n'y a pas de méthode polynomiale exacte pour construire un arbre automatiquement, seuls les algorithmes génétiques apportent une solution acceptable en termes de précision et de temps de calcul. Les frontières de décision sont strictes et provoquent des aberrations lorsque les données sont trop bruitées. La construction d'un arbre pour un problème donné n'est pas unique, l'existence d'un arbre optimal pour un problème donné n'implique pas qu'il soit équilibré et peu profond, la reconnaissance peut alors être très lente.

TABLEAU 1.8 – Résumé d'une vue d'ensemble de plusieurs algorithmes d'analyse de la structure logique. Le tableau présente la base de données utilisée, la métrique de performance, les résultats obtenus ainsi que l'idée clé de l'algorithme

Authors	Experimental Dataset	Performance Metric	Performance Results	Key Idea	
[Tsujimoto et Asada, 1990]	106 pages from various sources	N/S	94/106 accuracy	mapping a physical tree to a logical one	
[Yamashita et coll., 1991]	77 Japanese patent application front pages	cost function	59/77 accuracy	top-down layout model and relaxation labeling	
[Kreich et coll., 1991]	one page	confidence measure	N/S	knowledge based analysis	
[Fisher, 1991]	one page	N/S	N/S	rule-based	
[Derrien-Péden, 1991]	none	N/S	N/S	frame and macro-typographical based	
[Ingold et Armangil, 1991]	none	N/S	N/S	rule based, physical zones available	
[Brugger et coll., 1997]	five memo pages - one for training, four for testing	N/S	N/S	N -gram model, physical zones available	
[Conway, 1993]	none	N/S	N/S	page grammar	
[Krishnamoorthy et coll., 1993]	21 IBM journal pages for training, 12 IBM PAMI pages for testing	% area labeled, missed labels	reported for each of 12 IBM journal and IEEE PAMI pages	page parsing, block grammar	
[Saitoh et coll., 1993]	393 Japanese/ English pages for testing	six criteria based on result usage	results reported based on three criteria	text area influence rules	
[Tateisi et Itoh, 1994]	70 Japanese pages from books/magazines	N/S	87% and 82% logical labeling accuracy for manuals and technical papers etc.	stochastic grammars, physical zones available	
[Niyogi et Srihari, 1995]	44 newspaper pages	block classification, block grouping, read order accuracy	reported for each, of 32 newspaper pages and read order accuracy	rule-based, knowledge-based	
[Summers, 1995a]	196 pages from technical reports with corrected segmentation	Precise and generalized accuracy	85.5% logical labeling accuracy	logical prototype, matching, physical zones available	
[Dengel et Dubiel, 1996]	40 letters for learning, 40 letters for testing	recall, precision, F value	reported for 40 letters	logical structure learning, physical zones available	
[Lin et coll., 1997]	235 book pages	two types of errors, identification rate	reported for 235 pages	OCR and rule based	
[Ishitani, 1999]	150 pages from various sources	N/S	96.3% logical object extraction accuracy	emergent computation, rule based	
[Srihari et coll., 1999]	US postal address directory	N/S	ZIP code, city name state, street name	Shannon entropy	
[Kim et coll., 2001]	over 11,000 pages from over 1,000 biomedical journals	labeling	96.7% labeling accuracy	OCR and rule based	
[Chenevoy et Belaïd, 1991]	scientific articles	N/S	N/S	blackboard, learning, hypothesis managment	
[Akindele et Belaïd, 1995]	scientific articles	N/S	generated rules correspond to those obtained manually	learn the generic model, inference tree grammars	

TABLEAU 1.9 – Résumé détaillé de plusieurs algorithmes d'analyse de la structure logique. Le tableau indique si une analyse de l'erreur a lieu, la représentation des structures physique et logique ainsi que des sorties, les étiquettes logiques et le domaine d'application

	Authors	EA	Physical Layout Representation	Logical Structure Representation	Output Rep.	Logical Labels	Application Domain
	[Tsujimoto et Asada, 1990]	yes	block dominating rules, tree	tree	N/S	title, abstract, sub-title, paragraph, header, footer page number, caption	various document
	[Yamashita et coll., 1991]	yes	tree	tree	ODA	title, author, affiliation, body column, block	patent applications
	[Kreich et coll., 1991]	no	document style parameters	logical labels	N/S	sender, date, reference	N/S
	[Fisher, 1991]	no	rules, tree	rules, labeling	MIF	section heading, figure, figure caption, page heading, page footings	N/S
	[Derrien-Péden, 1991]	no	tree	rules, labeling	MML	title, list, paragraph abstract	N/S
	[Ingold et Armangil, 1991]	no	none	EBNF grammars, presentation rules	N/S	title, paragraph, section, chapter	N/S
	[Brugger et coll., 1997]	no	none	tree	N/S	N/S	memo pages
	[Conway, 1993]	no	page grammars	context-free string grammar	SGML	title, heading, paragraph, figure	N/S
	[Krishnamoorthy et coll., 1993]	no	block grammar, tree	block grammar, tree	N/S	title, author, abstract	journal pages
	[Saitoh et coll., 1993]	no	document style parameters	tree	N/S	body, caption, header footer	various documents
	[Tateisi et Itoh, 1994]	yes	none	grammar rules	N/S	headings, paragraph, list item	N/S
	[Niyogi et Srihari, 1995]	yes	rules	rules, tree	N/S	title, story, sub-story, photo, caption, graph	newspaper pages
	[Summers, 1995a]	no	none	logical prototypes	N/S	paragraph, heading, list item	technical reports
	[Dengel et Dubiel, 1996]	no	none	GTree	N/S	sender, recipient, date logo, subject, footer body-text	letters
	[Lin et coll., 1997]	yes	document style parameters	logical labels	N/S	headline, content, figure, table, page number, head-foot	book pages
	[Ishitani, 1999]	no	document style parameters	logical labels	N/S	headline, header, footer note, caption, program, formula, title, list	various documents
	[Srihari et coll., 1999]	no	information, uncertainty, redundancy	N/S	N/S	Zip code confirmation	Postal address interpretation
	[Kim et coll., 2001]	no	zones	logical labels	database tables	title, author affiliation, abstract	biomedical journals
	[Chenevoy et Belaïd, 1991]	no	none	EBNF grammars	ODA	N/S	scientific articles
	[Akindele et Belaïd, 1995]	no	ODA-like	set of rules	ODA	N/S	scientific articles

Les méthodes à apprentissage résolvent le problème de l'expert devant transformer les connaissances en règles pour le système de reconnaissance. En laissant le système lui-même s'adapter aux données d'apprentissage il sera capable de traiter avec plus de facilité des informations physiques erronées lors de la reconnaissance. Dans la pratique, les systèmes dirigés par le modèle mais à apprentissage se focaliseront plus sur la construction du modèle ou sur l'intégration de règles que sur les liens reliant les structures physiques et logiques. Les problèmes de convergence et de cas d'arrêt sont assez peu discutés et rajoutent en contrepartie de nouvelles limitations.

1.7 Conclusion

Les avancées dans le domaine la reconnaissance de documents sont certes effectives mais les méthodes utilisées ont encore en charge un certain nombre de difficultés lorsque l'extraction des indices physiques n'est pas de bonne qualité. Les plus citées utilisent des approches basées sur le modèle et entreprennent, pour couvrir la forte variabilité des documents, multiplier le nombre de règles pour les systèmes les plus simples ou multiplier le nombre de règles de productions pour les systèmes à base de grammaires.

Les méthodes dirigées par le modèle sont dépendantes d'un expert qui a l'obligation de formaliser les relations entre les observations physiques et les interprétations logiques correspondantes. Elles sont aussi sensibles à la qualité des observations mais également au changement de classe de document ; une modification même mineure au niveau de la classe de document peut entraîner une baisse des résultats de reconnaissance pour les moins flexibles d'entre elles. Elles ne proposent d'ailleurs pas toujours un score de confiance sur les résultats et un traitement du rejet est donc plus difficile à entreprendre.

Pour pallier les problèmes d'adaptabilité entre d'une part un modèle générique de document et d'autre part les données bruitées provenant de l'analyse physique, certains auteurs utilisent un apprentissage au sein de la méthode. Les moins avancées se limitent à une interactivité lors de la création de fonds de vérité ou lors de la correction après la reconnaissance. Celles utilisant réellement un apprentissage le font bien souvent indirectement sur le document car l'apprentissage sert à trouver les règles ou fixer les paramètres d'un système de type dirigé par le modèle. D'après notre connaissance, aucun système exclusivement dirigé par les données ne semble s'être imposé pour l'analyse de la structure logique alors que ce type d'approche est largement employé lors de l'analyse de la structure physique.

Nous avons aussi remarqué une très faible utilisation des méthodes neuronales qui sont pourtant capables d'apprentissage et très présentes dans les étapes précédant l'analyse logique. Leur absence doit s'expliquer en partie par le fait qu'elles se prêtent mieux à des problèmes provenant du traitement du signal et que les méthodes classiquement employées ne sont pas spécialement conçues pour traiter des données structurées [Marinai et coll., 2005]. La construction de documents de vérité est une tâche longue et fastidieuse et doit se faire au moins en partie manuellement. Il est donc compréhensible que les solutions dirigées par le modèle soient privilégiées car il semble plus naturel de vouloir utiliser un processus inverse à celui qui a permis la synthèse du document et qu'il est d'autant plus simple d'écrire dans ces formalismes des connaissances a priori.

Le but que nous nous sommes fixés pour la thèse est l'élaboration d'une méthode autonome capable d'établir seule les relations entre les observations de la structure physique et les éléments de la structure logique. Au vu des travaux déjà menés, il ne nous paraît pas pertinent d'utiliser une méthode reposant exclusivement sur une approche dirigée par le modèle. Il est cependant

utile et certainement nécessaire de conserver les atouts de cette dernière comme l'intégration de connaissances a priori. Il ne sera donc pas possible d'utiliser une méthode neuronale classique pour résoudre le problème.

Avant de détailler les fondements de notre méthode, nous allons montrer au cours du chapitre suivant l'intérêt et les capacités des méthodes neuronales en nous intéressant particulièrement au système à représentation locale de [Côté, 1997] qui permet l'intégration de concepts et effectue une reconnaissance perceptive des formes ainsi qu'au Perceptron multicouche qui nous permettra d'étendre le réseau des précédents auteurs à un fonctionnement plus dirigé par les données.

Chapitre 2

Réseaux de neurones à représentation locale et utilisation du contexte

Les systèmes de reconnaissance de structures logiques de documents proposés par la littérature sont principalement dirigés par un modèle et ne sont pas suffisamment flexibles et génériques pour traiter des images de documents complexes. Bien qu'ils tentent de reproduire une activité mentale de lecture, aucun n'utilise une véritable modélisation cognitive. En partant de constatations de psychologues sur les modèles de lecture [McClelland et Rumelhart, 1981] et d'une implémentation par [Côté, 1997], nous verrons comment étendre un modèle cognitif de lecture de mots à la reconnaissance de structures logiques de documents.

Sommaire

2.1 Introduction 31
2.2 Modèles cognitifs de lecture 32
2.3 Systèmes de lecture basés sur des principes cognitifs 33
2.3.1 Le système Perceptro 34
2.3.2 Réseau de neurones transparent 42
2.4 Le Perceptron multicouche 43
2.4.1 Le neurone 43
2.4.2 Topologie en couches 44
2.4.3 Apprentissage 46
2.4.4 Applications 49
2.5 Conclusion 50

2.1 Introduction

Les approches de reconnaissance de structures logiques de documents vues jusqu'à présent sont généralement fondées sur des systèmes figés, à base de règles ou de grammaires. De la connaissance est introduite par un expert et l'étape d'analyse repose directement sur ces informations. Bien qu'ils tentent de reproduire une activité mentale humaine, aucun n'emploie une véritable modélisation cognitive ni une approche perceptive lors de la reconnaissance qui apporterait un meilleur jugement sur la reconnaissance.

Cette orientation a pourtant déjà été expérimentée dans le cas de l'écriture manuscrite. Notre motivation à vouloir employer une approche cognitive nous permettra d'acquérir d'une part des moyens capables de surmonter les limitations des systèmes conventionnels et d'autre part, de les faire évoluer vers de véritables outils de reconnaissance s'adaptant aux diverses variations des documents.

Le système que nous proposons s'inspire d'approches basées sur des principes cognitifs empruntés aux travaux de [McClelland et Rumelhart, 1981] dans lesquels les auteurs cherchent essentiellement à imiter le comportement humain, à modéliser des stratégies adaptées qu'ils emploient pour faire coopérer différents niveaux d'interprétation, afin d'améliorer les performances de reconnaissance. Bien que nous nous soyons essentiellement inspirés des publications de [Côté et coll., 1998 ; Côté, 1997], d'autres références ont travaillé sur des principes cognitifs similaires et l'utilisation du contexte.

2.2 Modèles cognitifs de lecture

L'étude de modèles cognitifs nous a permis d'acquérir les explications sur la façon dont un lecteur humain fait coopérer ses connaissances afin d'adapter sa reconnaissance aux différentes variations que peut prendre une forme.

Classiquement, les travaux en psycholinguistique considèrent trois niveaux de traitement de l'information linguistique : le mot, la phrase et le texte. Les contributions sont largement plus nombreuses pour le plus bas niveau. En effet, l'identification des mots constitue une phase clé des processus impliqués dans la lecture ; elle est souvent l'étape préalable et indispensable aux systèmes traitant les informations à des niveaux plus élevés. Le mot représente l'unité de base du langage écrit, une étape charnière entre les processus de perception de bas niveau et les processus cognitifs de haut niveau.

La lecture est une activité qui met en jeu de nombreux niveaux de traitement de l'information, allant de la perception du mot jusqu'aux phénomènes complexes engagés dans la compréhension. Chaque niveau nécessite des investigations précises. Les processus entrant en jeu sont généralement si rapides et si automatiques que nous ne sommes pas conscients des étapes intermédiaires entre le moment où les mots sont projetés sur notre rétine et le moment où nous en comprenons le sens [Segui, 1991]. Cela dissimule une complexité et rend d'autant plus difficile leur abord expérimental.

Il existe cependant une chronologie d'événements dans le processus qui commence par l'extraction des informations dans la page et qui se termine par la compréhension du document. Entre ces deux événements on peut distinguer, suivant les auteurs, trois grandes étapes pour la lecture : l'identification lexicale, l'analyse syntaxique et le calcul sémantique. Dans [Baccino et Colé, 1995] le phénomène est apparenté dans sa globalité à un système de traitement de l'information (Fig. 2.1).

Différents *processeurs* sont impliqués dans cette chaîne comme ceux permettant la reconnaissance des formes qui détaillent les différentes étapes de transformation de l'information. Le rôle des diverses mémoires (différenciées par leur durée de persistance et par le type d'information conservée) est de faciliter le traitement en cours et de conserver à un moment donné le résultat des traitements.

Le mot constitue le point de convergence entre les différents niveaux de représentation tels que les niveaux visuels, orthographiques, lexicaux, syntaxiques et sémantiques supposés intervenir dans le traitement du langage écrit. En effet, la lecture d'un mot est une étape clé qui prend son importance du fait qu'elle permet l'accès au lexique mental [Taft, 1991].

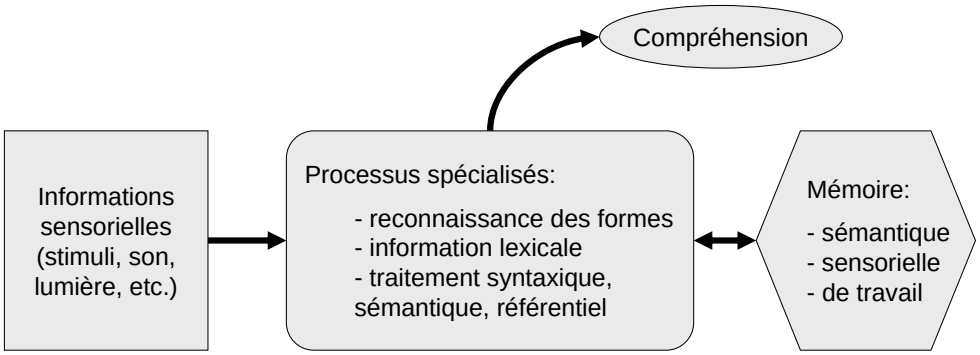


FIGURE 2.1 – Représentation d'un système humain de traitement de l'information appliqué au processus de lecture selon [Baccino et Colé, 1995]

De l'ensemble de ces considérations, ressort un point très important, largement développé par Côté, qui est la prise en compte du «contexte» que nous retiendrons pour notre système. Il est en effet difficile de parler des modèles cognitifs sans évoquer l'importance du contexte et des effets contextuels observés chez l'homme dans la reconnaissance des formes. La pertinence d'un modèle cognitif est d'ailleurs justement évaluée en fonction de son aptitude à rendre compte des différents effets contextuels observés chez l'homme. Dans le cadre de la reconnaissance des mots, les travaux dont nous nous sommes inspirés développent plus précisément l'effet de la supériorité du mot qui joue dans ce domaine le rôle de contexte lexical [Reicher, 1969].

2.3 Systèmes de lecture basés sur des principes cognitifs

L'originalité des travaux de Côté vient de l'utilisation d'un modèle neuronal, d'autres mises en œuvre sont possibles partageant les mêmes idées clés. [Bramall et Higgins, 1995] proposent par exemple une approche de reconnaissance des mots inspirée du modèle des logogènes proposé par [Morton, 1969] et implémentée par une architecture de type tableau noir. Les données sont organisées de façon hiérarchique partant des informations bas niveau qui décrivent les mouvements du stylo jusqu'aux plus hautes désignant des connaissances lexicales. Les sources de connaissance sont organisées également de façon hiérarchique et correspondent à une caractéristique particulière. Pour reconnaître un mot, trois opérations de filtrage du lexique se succèdent : une génération des mots hypothétiques, une réduction d'ambiguïté générale entre les mots candidats et une phase d'élimination d'ambiguïtés spécifiques travaillant sur un nombre limité de mots partageant des caractéristiques similaires. Le résultat final de cette phase est une liste de solutions mots. [Pasquer et coll., 2000] s'inspirent d'un modèle d'interprétation multi-contextuelle pour la reconnaissance de l'écriture en ligne. Il est emprunté à [Anquetil, 1997] qui utilisait le modèle dans le cadre de la reconnaissance de mots par logique floue, lui-même inspiré du modèle d'activation interactive de McClelland et Rumelhart. Les auteurs proposent aussi une organisation hiérarchique des niveaux de traitement des informations extraites de l'image du mot. Deux principes sont mis en œuvre : une organisation hiérarchique des informations extraites sur l'image et un processus interactif de circulation des informations entre quatre niveaux (les modèles de lettres hors contexte, les bigrammes de modèles de lettres, les bigrammes de lettres et enfin les mots).

2.3.1 Le système Perceptro

Le travail de thèse de Côté se concentre sur une réponse générale au problème de la lecture automatique de l'écriture cursive ; les idées clés développées étant l'adoption de modèles de lecture et l'exploitation d'informations contextuelles afin d'imiter au mieux l'habilité humaine à lire.

Modèles de lecture et segmentation

Les modèles de lecture sont le résultat d'investigations de différents domaines comme la biologie, la neurophysiologie, la psychologie cognitive ou bien encore la linguistique [Taylor et Taylor, 1983]. La plupart des travaux menés sur la lecture se concentrent sur l'écriture imprimée, mais il a été démontré que moyennant une «normalisation», les mécanismes de lecture se transposent de manière similaire sur le cursif.

L'une des difficultés supplémentaires dans le cadre des travaux de Côté vient du fait que la reconnaissance se fait hors ligne. Le scripteur étant absent, il ne reste plus que l'image de son tracé. Le signal est donc uniquement bidimensionnel (une matrice de pixels), l'information temporelle étant perdue. Le système développé, nommé Perceptro, débute son analyse de l'image numérisée d'un mot cursif et identifie ce mot parmi une liste de mots candidats potentiels.

Le premier problème à résoudre est celui de la segmentation du mot en lettres (Fig. 2.2). Quel que soit le modèle de lecture, tous se servent, à un stade ou à un autre, d'une segmentation même implicite du mot en lettres. Ce problème de segmentation se retrouvera aussi inévitablement dans nos travaux (segmentation de l'image entière en blocs de textes homogènes) et nécessitera aussi un traitement particulier. Nous nous retrouvons aussi confrontés au paradoxe de Sayre *segmenter pour reconnaître et reconnaître pour segmenter*. Côté, tout comme la majorité des méthodes actuelles, pratique une approche hybride qui alterne approche analytique et approche globale [Casey et Lecolinet, 1996]. Elle consiste à émettre des hypothèses sur le mot à reconnaître juste après une première présegmentation puis de valider ou corriger la segmentation.

FIGURE 2.2 – Mot manuscrit cursif et problème de segmentation

Réseau à représentation locale

Tout comme dans la reconnaissance de la structure logique, les modèles connexionnistes comme le Perceptron multicouche (PMC) trouvent très peu d'applications dans l'écriture manuscrite au dépend principalement de méthodes à base de HMM [Steinherz et coll., 1999]. Ils ne

sont réellement présents que dans des travaux traitant de lettres ou de chiffres isolés [Martin et Pittman, 1989 ; Le Cun et coll., 1990a].

Côté a fait le choix de prendre comme point de départ le modèle connexionniste de McClelland et Rumelhart pour son système. Elle doit résoudre de ce fait les deux principaux problèmes inhérents à ce type de méthode : d'une part la complexité en temps de calcul et le grand nombre d'échantillons nécessaires à l'apprentissage, d'autre part la difficulté d'explication du comportement de ce type de réseau qui s'avère être un élément bloquant pour la modélisation de la perception humaine. Partant de différentes constatations sur les stratégies de reconnaissance, de segmentation et de travaux sur la lecture, le système Perceptro utilise une architecture singulière : elle est connexionniste à représentation locale. Chaque neurone représente un concept contrairement aux réseaux à représentation distribuée, tels que les PMC, dans lesquels certains neurones n'ont pas de signification intrinsèque. Le choix est justifié pour plusieurs raisons :

- il est possible d'intégrer dans le réseau des connaissances a priori ;
- l'explication du réseau pas à pas est plus aisée ;
- l'architecture permet de décomposer l'analyse rendant l'approche moins globale comme le ferait un réseau à représentation distribuée ;
- les poids ne sont pas modifiés par apprentissage réduisant la taille des bases de données ainsi que les temps de calcul ;
- l'information est contextuelle, correspondant à des fondements spécifiques à la lecture des mots. À savoir que dans leur cadre applicatif, la reconnaissance de toutes les lettres n'est pas nécessaire pour reconnaître le mot et la lecture du mot ne s'effectue pas uniquement de gauche à droite mais combine à la fois une approche ascendante et descendante.

Plusieurs modèles de lecture sont possibles, la plupart simule l'accès lexical, processus par lequel l'image du mot est associée à une signification par le cerveau humain comme le *verification model* de [Becker, 1976] et le *dual route* de [Coltheart et Rastle, 1994]. Le modèle de McClelland et Rumelhart a été retenu car il s'approche le plus de la perception humaine et permet dans le même temps d'utiliser du contexte pour améliorer la segmentation de part l'intégration d'informations contextuelles.

Le contexte ici représente le phénomène « d'effet de la supériorité du mot » qui se manifeste par un temps de reconnaissance plus court d'une lettre dans un mot, temps qui se trouve être plus long lorsque la lettre est montrée isolément du mot (Fig. 2.3). Il en est de même pour les pseudo mots ou syllabes mais le phénomène ne peut pas être généralisé pour n'importe quel regroupement de lettres. Cette dernière remarque implique que le contexte ne peut être défini empiriquement mais se doit d'être construit de manière logique. Le modèle de lecture proposé est transposé au réseau connexionniste (Fig. 2.4).

Le modèle de McClelland et Rumelhart décompose la reconnaissance en plusieurs paliers qui représentent chacun des niveaux d'abstraction différents. Dans le cadre de la reconnaissance du cursif, le modèle en comporte trois : les primitives, les lettres et pour finir les mots, le tout formant une structure hiérarchique (Fig. 2.5). Chaque élément d'une couche est associé à un neurone ; ainsi la couche de lettres comprend par exemple 26 neurones. Les connexions entre les neurones de couches différentes peuvent être excitatrices ou inhibitrices et il existe également des connexions au sein d'une même couche de neurones mais elles sont alors toutes inhibitrices entre elles (ex : connexions inhibitrices entre les neurones mots).

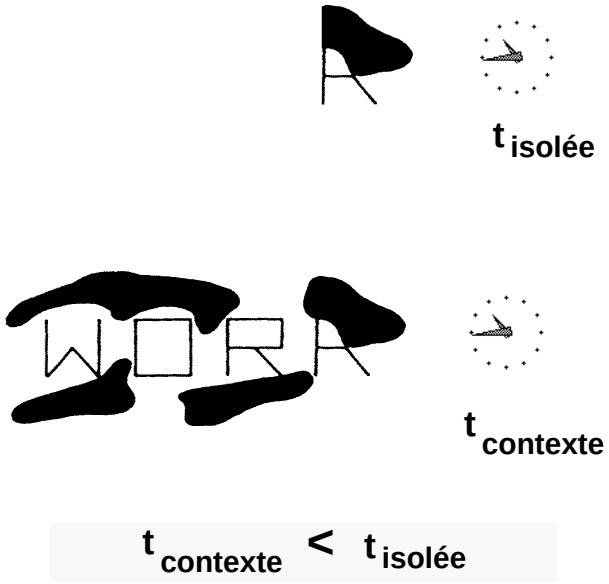


FIGURE 2.3 – Observations expérimentales de la supériorité du mot sur la lettre isolée

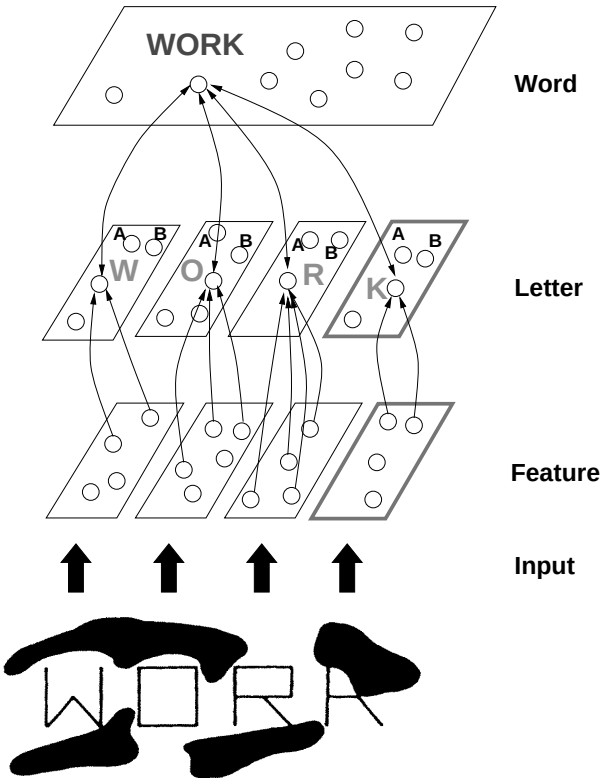


FIGURE 2.4 – Modèle d'activation interactive

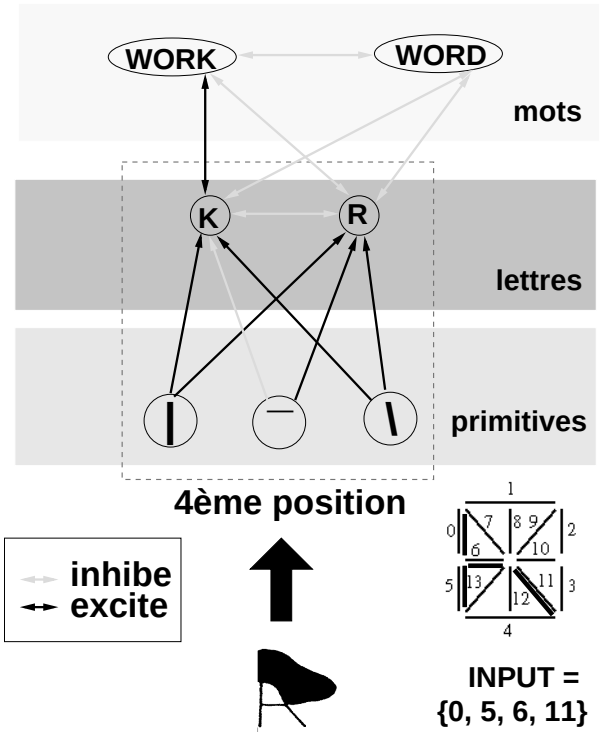


FIGURE 2.5 – Types de connexions entre les niveaux et les neurones selon [McClelland et Rumelhart, 1981]

Caractéristiques du réseau

Plusieurs règles sont définies pour faire fonctionner le réseau :

- l'activation est le cumul des énergies des lettres vues dans le mot et de la perception des lettres prises isolément ;
- la fonction d'activation utilisée est une sigmoïde en raison de ses similitudes avec le neurone biologique ;
- l'interaction se fait à l'intérieur du réseau : entre chaque couche, les connexions sont excitatrices et réciproques. Les décisions prises à chaque niveau sont ainsi dépendantes de l'influence de tous les autres niveaux ;
- les connexions inhibitrices et réciproques entre les neurones permettent une interaction de type compétitif entre eux.

Le mécanisme de reconnaissance se déroule de manière naturelle : à l'introduction d'un mot dans le réseau, des neurones primitives sont activés lorsqu'ils rencontrent une primitive connue. Le stimulus est propagé dans les paliers supérieurs jusqu'au niveau des mots puis une rétroaction des mots vers les lettres s'effectue pour renforcer le stimulus initial. On peut dès lors exploiter l'information donnée par les neurones mots (le contexte) pour reconnaître plus facilement les lettres et lever les ambiguïtés. Il est à noter que dans ce modèle, les connexions peuvent être

inhibitrices et excitatrices. Dans les systèmes de Côté et de Snoussi Maddouri, que nous verrons plus tard, les auteurs ont fait le choix de ne garder que le mécanisme excitateur. Celui de McClelland et Rumelhart permet la coopération de part les liens excitateurs : si un neurone est stimulé, il entraîne avec lui les autres neurones qui lui sont connectés. La compétition se traduit par l'accentuation des différences initiales entre deux neurones lorsqu'ils sont connectés par des liens inhibiteurs.

En contrepartie, le système Perceptro intègre d'autres éléments qui correspondent mieux à des spécificités du problème étudié que ceux proposés par [McClelland et Rumelhart, 1981]. Tout comme nous le ferons dans nos travaux sur la reconnaissance de structures logiques, il est important de tenir compte de la variabilité des entrées. Le cursif, ainsi que la structure logique, sont soumis à des règles générales et communes mais chaque scripteur ou éditeur a sa propre manière de formuler ses conventions. Mis à part les rares cas spécifiques où le tracé est normalisé (ex. : dessin industriel) ou bien encore des documents extrêmement formatés (ex. : formulaires), il faut toujours prendre en considération la nature imparfaite des données du problème à traiter. Cette variabilité est encore plus accentuée par les outils d'extraction qui donneront les primitives d'entrée et qui eux aussi sont souvent imprécis, avec une qualité ou taux d'erreur inquantifiable.

Spécificités du système Perceptro

Pour revenir au problème du cursif, Côté propose plusieurs changements dont les trois principaux sont :

- l'utilisation de primitives plus spécifiques au cursif ;
- l'abandon de l'inhibition pour éviter de perdre de l'information trop rapidement et risquer d'oublier des hypothèses ;
- la position des lettres dans le mot n'est plus figée mais utilise la notion de flou afin de traiter les ambiguïtés locales.

Contrairement à d'autres modèles neuronaux qu'il est commun de retrouver dans la littérature, le système Perceptro ne dispose pas d'apprentissage : les poids des connexions sont fixés par de la connaissance a priori. Deux lexiques permettent de relier et de pondérer les neurones d'une couche à l'autre : un lexique primitives↔lettres et un lexique lettres↔mots. Pour chacun des mots du lexique, un tableau d'étiquetage relie de manière dynamique chacune des lettres du mot avec les zones correspondantes dans l'image. Ce choix n'est pas réellement justifié par les auteurs et ne dépend malheureusement que de considérations statistiques. Plutôt que de partir d'un mécanisme d'apprentissage qui déterminerait de manière optimale les liens entre chaque neurone, le choix a été de créer autant de neurones qu'il y a de possibilités d'arrangement entre les neurones mots, leurs lettres et leurs primitives. Si m est le nombre de mots dans le lexique, l la longueur maximale d'un mot et p le nombre de primitives, il y a au total $m \times l \times p \times 26$ neurones. Selon les auteurs, cette méthode ne perturbe en rien les résultats et aurait l'avantage certain d'être beaucoup plus rapide qu'un apprentissage classique.

Cycles perceptifs

Le réseau du système Perceptro effectue, comme énoncé précédemment, plusieurs passages entre les entrées et les sorties et ceci dans les deux sens. La première phase qualifie un processus dit ascendant où, comme dans un réseau classique, l'information des neurones d'entrée (ou primitives) est propagée dans toutes les couches supérieures jusqu'aux sorties. Cette phase achevée, certains neurones présents sur chaque couche deviennent actifs. Les plus intéressants

se trouvent sur la couche finale qui contient les mots du lexique et permet déjà d'identifier plusieurs solutions possibles. Là où s'arrêterait un réseau classique comme le PMC, il existe ici un processus descendant qui consiste à propager en sens inverse l'activation de la sortie jusqu'aux entrées. C'est justement pendant ce processus que l'information de contexte (ou la «supériorité du mot») est prise en compte. L'activation de certains neurones mots donne des indices sur l'identité des lettres inconnues présentes dans l'image. De même l'information des lettres même erronées peut générer des hypothèses sur l'exactitude des primitives extraites (Fig. 2.6).

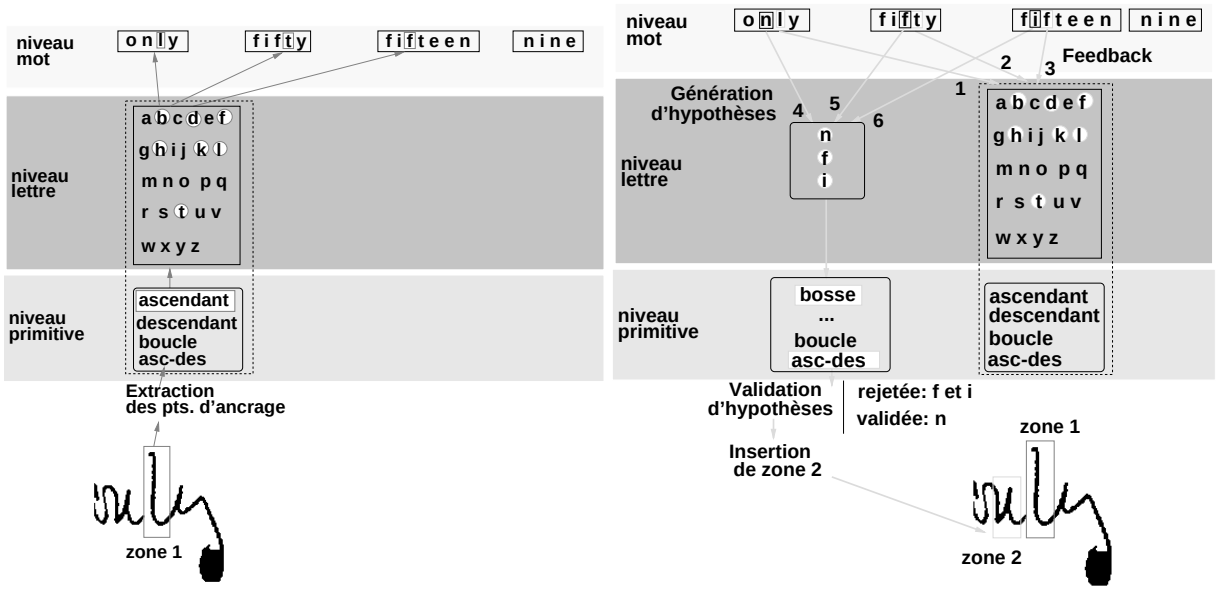


FIGURE 2.6 – Propagation de l'activation chez Côté : à gauche un processus ascendant, à droite un processus descendant

La succession d'un processus ascendant et d'un processus descendant constitue un cycle perceptif. Les neurones du système ont besoin de plusieurs cycles perceptifs avant d'atteindre un niveau d'activation suffisamment important pour décider de l'étiquette d'un mot à reconnaître. Ainsi, à la fin d'un cycle perceptif, des hypothèses sont validées et les détecteurs de primitives qui leur correspondent sont activés. Ces nouvelles primitives activées sont ajoutées à celles déjà présentes au commencement du processus.

Comme l'activation est uniquement croissante, les neurones arrivent à saturation au bout d'un certain nombre de cycles. Dès qu'un neurone mot a atteint sa valeur maximale, on en déduit que le système a convergé vers une solution et ce neurone sera le résultat de la reconnaissance.

Fonction d'activation

L'équation d'activation est différente dans les travaux de McClelland et de Côté. Pour ces premiers, la fonction A s'écrit de manière itérative en faisant la différence entre une contribution des voisins excitatrice ou inhibitrice :

$$A_i(t + \delta t) = A_i(t) - \theta_i(A_i(t) - r_i) + E_i(t) \tag{2.1}$$

avec $n_i(t)$ la somme des excitations et des inhibitions sur le neurone i :

$$n_i(t) = \sum_j \alpha_{ij} a_j(t) - \sum_j \beta_{ij} a_j(t) \tag{2.2}$$

où α_{ij} et β_{ij} représentent respectivement les poids pour l'excitation et l'inhibition entre le neurone j et le neurone i, $a_j(t)$ l'activation au temps t du neurone j voisin du neurone i. La contribution des voisins $E_i(t)$ est définie en fonction de $n_i(t)$:

$$E_i(t) = \begin{cases} n_i(t)(M - A_i(t)) & n_i(t) > 0 \\ n_i(t)(A_i(t) - m) & n_i(t) < 0 \end{cases} \tag{2.3}$$

où M et m sont les bornes respectivement supérieure et inférieure de l'activation (fixées empiriquement à 1 et -0,02). Dans la formule de l'activation du neurone i à l'instant $t + \delta t$, θ_i est une constante de décroissance de l'activation du neurone i et r_i le niveau de repos de la cellule.

Pour le système Perceptro, l'inhibition n'est pas considérée, la contribution des voisins est donc modifiée. En développant la formule de l'activation, on obtient les différences présentées dans le tableau 2.1.

Activation pour [McClelland et Rumelhart, 1981]	Activation pour [Côté, 1997]
$n_i(t) < 0$, $A_i(t+\delta t) = (1-\theta_i)A_i(t) + \theta_i r_i + n_i(t)(M - A_i(t))$ $n_i(t) > 0$, $A_i(t+\delta t) = (1-\theta_i)A_i(t) + \theta_i r_i + n_i(t)(A_i(t) - m)$ avec $n_i(t) = \sum_j \alpha_{ij} a_j(t) - \sum_j \beta_{ij} a_j(t)$	$n_i(t) > 0$ dans tous les cas $A_i(t + \delta t) = (1 - \theta_i)A_i(t) + n_i(t)(M - A_i(t))$ avec $n_i(t) = \alpha'_{ij} a(j)$

TABLEAU 2.1 – Comparaison des fonctions d'activation de [McClelland et Rumelhart, 1981] et de [Côté, 1997]

Détermination des poids du réseau

Les poids du système ne sont pas appris, ils s'adaptent durant le traitement en suivant les équations statistiques suivantes :

- $\alpha_{pl} = \frac{1}{NP}$ les poids entre la couche des primitives et la couche des lettres avec NP le nombre de primitives trouvées dans l'image pour la lettre l ;
- $\alpha_{lm} = \mathcal{F}(\Delta)_{lm} \frac{1}{NZ}$ les poids entre la couche des lettres et la couche des mots avec $\mathcal{F}(\Delta)_{lm}$ le coefficient-position (faisant intervenir un concept de position floue) de la lettre l dans le mot m et NZ le nombre de zones trouvées dans le signal à l'instant t ;
- $\alpha_{ml} = \frac{1}{NM}$ les poids entre la couche des mots et le lexique avec NM le nombre de mots dans le lexique contenant la lettre l.

Génération, validation et insertion d'hypothèses

L'exploitation du contexte se réalise suivant trois mécanismes successifs (Fig. 2.7). Le premier est la génération d'hypothèses dont la finalité est de trouver les lettres qui n'ont pas été encore reconnues. En fonction des informations déjà obtenues, plusieurs zones d'ancrage sont déterminées afin de tenter de découvrir la boîte englobante pouvant contenir une lettre du mot.

Le second mécanisme se charge de valider les hypothèses émises lors du précédent processus ; une lettre est validée si les primitives qui la décrivent peuvent effectivement être retrouvées dans l'image. Pour finir, le mécanisme d'insertion se charge de créer et d'insérer une zone parmi celles déjà trouvées pour chaque mot vraisemblable du lexique et chaque lettre validée de ce mot. L'utilisation de ces trois mécanismes permet d'apporter à la méthode une souplesse et les indications nécessaires permettant de réaliser une segmentation implicite du mot et d'effectuer des cycles perceptifs de plus en plus efficaces grâce à la rétroaction sur l'image d'entrée.

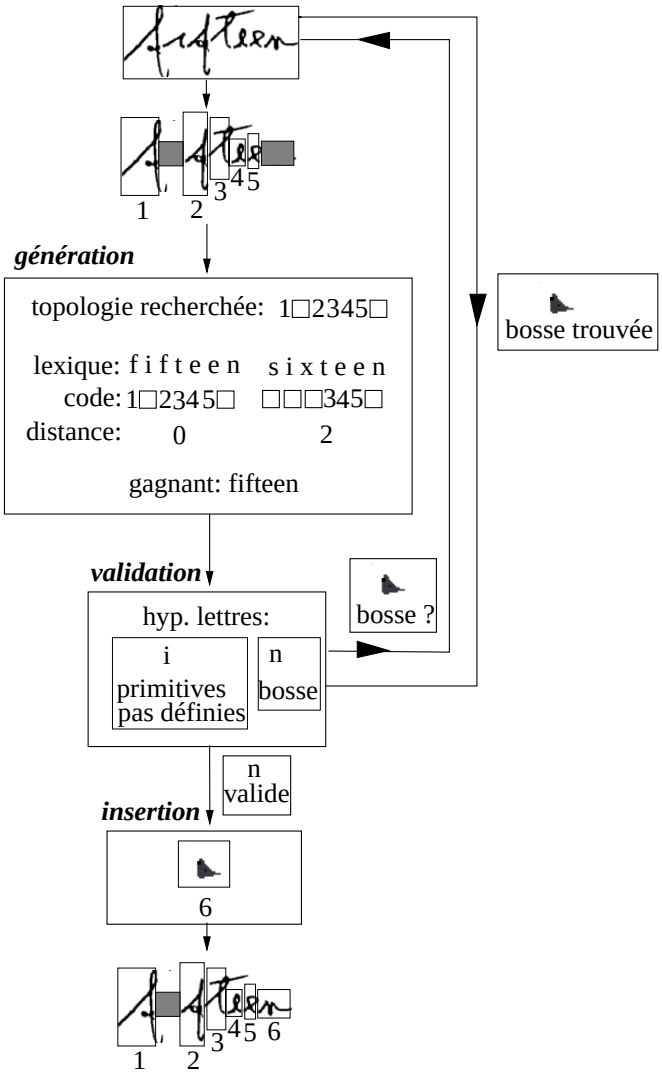


FIGURE 2.7 – Représentation de l'état interne du système Perceptro après un cycle ascendant

Testé sur une base de 3 000 images de mots manuscrits provenant de montant littéraux de chèques, le système Perceptro obtient un taux de reconnaissance moyen de 74% avec un taux maximum de 81% pour les mots de huit lettres.

2.3.2 Réseau de neurones transparent

Le système de Côté a été repris en partie dans notre équipe par [Snoussi Maddouri, 2003] qui l'a étendu au problème de la reconnaissance de mots manuscrits arabes. Le système est appelé réseau de neurones transparent (RNT) pour marquer son opposition avec les réseaux à représentation distribuée qualifiés de «boîtes noires». Pour s'adapter aux spécificités de l'arabe, leur réseau garde une topologie hiérarchique mais dispose de quatre couches au lieu de trois : une couche de primitives physiques, une couche de lettres, une couche supplémentaire de syllabes ou plutôt de PAW *(Piece of Arabic Word)* et pour finir, une couche contenant les mots du lexique (Fig. 2.8).

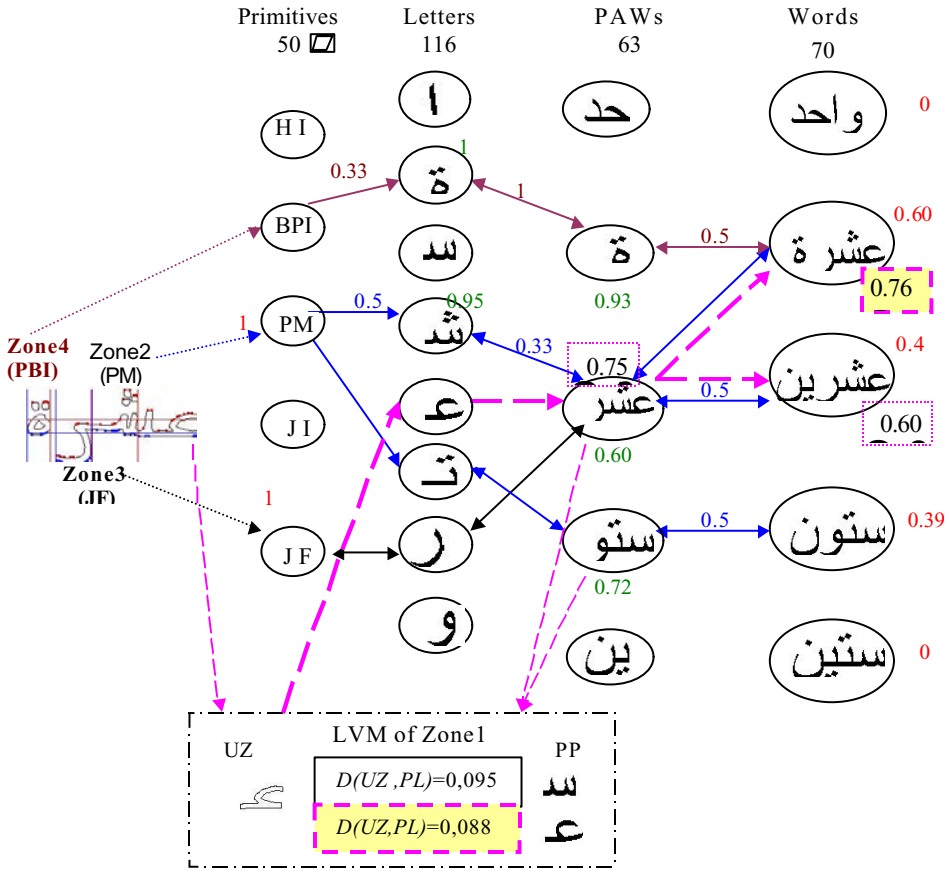


FIGURE 2.8 – Instance de réseau de neurones transparent proposé par Snoussi Maddouri pour la reconnaissance de l'écriture arabe manuscrite. L'hypothèse du mot permet de remonter les liens et de corriger la segmentation des lettres ou d'apporter de l'information supplémentaire par des descripteurs de Fourier

Tout comme le système Perceptro, le RNT fonctionne par propagation des activations de la couche des primitives jusqu'à la couche des mots et par rétropropagation contextuelle de la couche de mots à la couche des lettres. La propagation fournit une liste de mots candidats classés par activation décroissante pour permettre la reconnaissance. La rétropropagation, quant à elle, permet en cas d'ambiguïté, de trouver une correspondance entre les zones de lettres dans l'image du mot et les lettres du mot du lexique en utilisant l'information contextuelle. Les primitives d'entrée sont bien sûr différentes et spécifiques au script arabe.

Pour effectuer une meilleure correction des entrées, les auteurs proposent d'utiliser des descripteurs de Fourier [Snoussi Maddouri et coll., 2000] pour aider la correction des entrées. Plutôt que d'utiliser des descripteurs comme les hampes, les jambages, etc., l'idée est de considérer les coefficients de Fourier comme décrivant le contour d'un caractère et ainsi de créer une normalisation du caractère qui le rend plus robuste aux variations des scripteurs. En utilisant l'accumulation des harmoniques, il est possible de créer différentes visions d'une même lettre de la plus générale à la plus détaillée. Nous reprendrons un principe similaire (partitionnement de l'espace d'entrée) qui consiste à n'utiliser de l'information détaillée qu'en cas de nécessité.

L'utilisation des descripteurs de Fourier a été intégrée au RNT et le système a été testé sur des montants littéraux de chèques [Snoussi Maddouri et coll., 2002]. Le lexique contient 70 mots et 2100 images sont testées. Les résultats sans descripteurs sont de l'ordre de 90% et de 97% avec l'emploi des descripteurs.

Plus récemment, ces travaux ont été repris par [Bouriel et coll., 2005] qui proposent d'apporter un apprentissage au RNT. La méthode utilise un réseau ayant la même fonction d'activation que celle présentée dans le tableau 2.1 page 40. Les auteurs proposent d'apprendre, par l'algorithme de rétropropagation du gradient les poids d'un réseau de type PMC ayant la même topologie que le réseau transparent. Les poids trouvés sur le PMC sont ensuite disposés sur le RNT. La méthode ne nous paraît pas valide de part la présence d'une sigmoïde comme fonction d'activation pour le PMC et d'une fonction différente pour le RNT et de par le fait que les neurones des couches cachées ne peuvent pas s'assimiler aux neurones porteurs de concepts dans le RNT. Des taux d'environ 60% sont donnés sur la base IFN/ENIT.

2.4 Le Perceptron multicouche

Après avoir étudié le système à représentation locale Perceptro, nous nous proposons d'apporter des solutions aux faiblesses relevées dans l'approche. Afin de résoudre le problème de la détermination des poids et proposer une méthode plus dirigée par les données, nous allons employer certain principes d'un réseau de type Perceptron multicouche pour bénéficier entre autre de son apprentissage et de ses capacités de généralisation. Nous détaillerons son fonctionnement au cours de ce chapitre et nous illustrerons l'intérêt d'utiliser un tel classifieur au cœur de notre méthode [Cornuéjols et Miclet, 2002 ; Dreyfus et coll., 2002].

2.4.1 Le neurone

Le Perceptron de Rosenblatt [Rosenblatt, 1958] transpose le comportement des neurones en une équation intégrant les grands principes observés dans la nature. À partir d'un stimulus représenté par un vecteur d'observations $x \in \mathbb{R}^n$, chaque composante $x_i, i \in [\![1, n]\!]$ est multipliée par un poids de connexion w_i. La somme de ces entrées pondérées est ensuite faite en y ajoutant un biais θ (ou seuil d'activation). Pour des raisons de commodité de notation, on transforme ce biais en un nouveau neurone de sortie 1 avec un lien de poids w_0 tel que $\theta = w_0$. Pour finir, la somme (ou état d'activation) est passée dans une fonction f (ou fonction de seuil) non linéaire de type escalier. Au final la sortie d'un neurone s'écrit de manière synthétique :

$$y = f\left(\sum_{i=0}^{n} w_i x_i\right) \tag{2.4}$$

Seul, le Perceptron peut être vu comme une fonction de discrimination entre deux classes pour un problème de reconnaissance des formes : il partitionne l'espace d'entrée en deux régions avec une frontière de décision linéaire. Avec des poids w_i bien étudiés, cette surface linéaire permet de représenter des fonctions logiques comme le ET $\supset$, le OU $\supset$ et le NON $\rhd\!\!\circ$. Le Perceptron ne peut pas simuler le OU exclusif (XOR $\rotatebox{180}{$\supset$}$) car dans ce cas, la surface de décision est non linéaire. La critique du cas du XOR par [Minsky et Papert, 1969] a d'ailleurs provoqué une temporaire mais historique désaffection pour le Perceptron.

Bien qu'un neurone informatique ne soit pas une modélisation parfaite de sa version biologique, il n'en reste pas moins proche expérimentalement des phénomènes observés. De plus, agrégé en réseau, les capacités pourtant limitées du neurone artificiel produisent des résultats très intéressants tant au niveau purement fonctionnel qu'au niveau de la modélisation.

Plusieurs algorithmes ont été proposés pour déterminer les poids w_i, à commencer par la méthode de [Rosenblatt, 1958] à base d'expériences, la technique des moindres carrés [Widrow et Hoff, 1960] et enfin les techniques à base de descente de gradient comme dans le cas du Perceptron multicouche.

2.4.2 Topologie en couches

Si les unités élémentaires sont souvent très proches dans la plupart des systèmes neuronaux, c'est au niveau de l'agencement (ou architecture) de ces neurones que les systèmes se différencient. Comme évoqué dans la section précédente, les possibilités du Perceptron sont limitées. Même à plusieurs, ils ne peuvent délimiter que des régions aux frontières linéaires dans un espace de $\mathbb{R}^n$. En revanche, si les neurones sont placés en couches successives (les sorties d'un certain nombre de neurones sont les entrées des suivants et ainsi de suite jusqu'à la sortie), alors l'ensemble du réseau est capable de décider d'un problème pour des surfaces plus complexes et peut aussi simuler n'importe quelle fonction booléenne. Ce type d'organisation (Fig. 2.9) est appelé Perceptron multicouche (PMC).

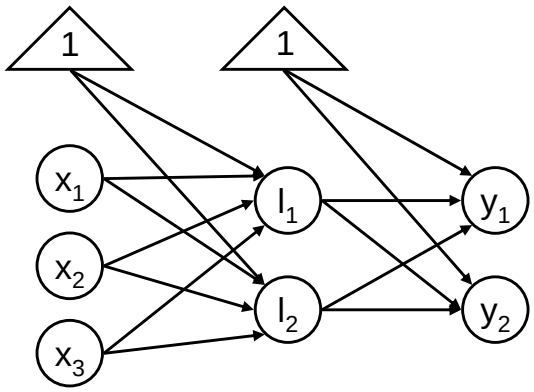


FIGURE 2.9 – Exemple de Perceptron multicouche. L'entrée est dans $\mathbb{R}^3$, la sortie dans $\mathbb{R}^2$. Il est complètement connecté et possède une couche cachée de deux neurones

Les unités de calcul ne sont plus appelées Perceptrons mais plus simplement neurones ou encore nœuds. Outre la topologie en couches, la principale différence avec la version de

[Rosenblatt, 1958] vient de l'utilisation de fonctions d'activation dérivables et non linéaires telles que la sigmoïde (Fig. C.1, p. 145), encore appelée fonction logistique, qui remplacent la fonction *sign*.

L'idée d'une telle topologie est ancienne et il a fallu attendre un certain nombre d'années pour voir apparaître des algorithmes permettant de calculer les poids d'un tel réseau en particulier à cause de l'introduction des couches cachées. Proposé pour la première fois par [Werbos, 1994] en 1974, l'utilisation de la rétropropagation du gradient de l'erreur dans des systèmes à plusieurs couches sera de nouveau mise au devant de la scène en 1986 par [Rumelhart et coll., 1986], et simultanément, sous une appellation voisine, chez [Le Cun, 1985] durant sa thèse.

Ces réseaux sont souvent totalement connectés, ce qui signifie que chaque neurone d'une couche i est connecté à tous les neurones de la couche $i+1$. Par contre, dans un schéma classique, les neurones d'une même couche ne sont jamais reliés entre eux.

Les PMC sont essentiellement employés à deux tâches : le partitionnement d'un espace de formes pour des problèmes de classification et l'approximation de fonctions. Contrairement au Perceptron de [Rosenblatt, 1958], le PMC peut représenter n'importe quelle fonction booléenne à n variables, bien que certaines puissent requérir un nombre exponentiel en n de neurones dans les couches cachées. Du fait de la non-linéarité de la sigmoïde comme fonction d'activation, les frontières de séparation s'adaptent mieux à chaque classe dans le cas d'un problème de classification. Cette propriété se retrouve aussi dans le cas de l'approximation de fonctions qui produit des courbes continues et lisses à la fois.

Les PMC possèdent des propriétés mathématiques intéressantes. Beaucoup d'entre-elles sont valables pour des réseaux à seulement deux couches cachées, ce qui témoigne de la puissance potentielle des PMC. Il est à noter que ces propriétés sont rarement constructives dans le sens où bien qu'il soit démontré qu'un certain nombre de neurones soit suffisant pour réaliser une tâche, la propriété ne donne aucune information sur la topologie à choisir afin de résoudre le problème (Annexe C.2, p. 145).

La majeure partie des propriétés sont prouvées sans l'hypothèse de l'utilisation de la sigmoïde, il suffit simplement que la fonction d'activation soit bornée (majorée et minorée), croissante et continue. On retrouve dans les implémentations, et suivant l'application, la tangente hyperbolique, la fonction erreur ou bien encore la fonction $x \rightarrow \frac{x}{1+|x|}$.

Le choix de la fonction se fait généralement sur des considérations relatives au temps de calcul. Certaines fonctions font converger lentement le réseau mais donnent une solution de bonne qualité après un grand nombre d'époques. D'autres font converger le réseau très rapidement vers une solution mais atteignent difficilement par la suite une meilleure solution même si le nombre d'époques est élevé. Si la partie « linéaire » de la courbe est trop courte et très pentue, on revient alors à une fonction escalier, ce qui peut conduire à des « sauts » lors de l'apprentissage. À l'inverse, une partie « linéaire » trop étalée et horizontale peut aboutir à une convergence extrêmement lente du réseau. Au vu des nombreuses expériences menées par la littérature, il semble que la sigmoïde soit la plus répandue dans les implémentations de PMC.

La difficulté d'utilisation de ce réseau réside dans le fait qu'il faille déterminer sa topologie ; il s'agit de définir le nombre de neurones des différentes couches ainsi que leurs interconnexions. Si le nombre de neurones cachés est trop faible, l'algorithme d'apprentissage n'arrivera pas à construire une représentation intermédiaire du problème qui soit linéairement séparable et certains des exemples ne seront pas appris correctement. Inversement, si ce nombre est trop élevé, il y a risque d'apprentissage par cœur du problème : le réseau reconnaît parfaitement les exemples d'apprentissage mais donnera des résultats médiocres sur des nouvelles données qu'il n'a pas vues durant l'apprentissage.

2.4.3 Apprentissage

L'approche la plus connue pour apprendre les poids d'un PMC est la technique de descente de gradient. En effet, l'utilisation de fonctions d'activation différenciables permet d'utiliser cette technique à la fois simple à mettre en œuvre et surtout très efficace sur le plan calculatoire.

Nous utiliserons dans la suite de ce chapitre les notations suivantes :

- P le nombre de formes dans la base d'apprentissage ;
- $x_p, p \in [\![1, P]\!]$ la forme n° p de la base d'apprentissage ;
- L le nombre de couches du réseau (y compris la couche d'entrée et de sortie) ;
- $N_l, l \in [\![0, L-1]\!]$ le nombre de neurones dans la couche n° l ;
- $o_{l,j}$ la sortie calculée du neurone n° j dans la couche n° l. On considère que $o_{l,0}$ contient le biais égal à 1 ;
- $d_j(x_p)$ la composante n° j de la sortie attendue pour la forme x_p ;
- $w_{l,j,i}$ le poids de la connexion entre le neurone n° i dans la couche $l-1$ et le neurone n° j dans la couche l ;
- f la fonction d'activation.

La sortie d'un neurone quelconque est donnée par :

$$o_{l,j} = f\left(\sum_{i=0}^{N_{l-1}} w_{l,j,i} o_{l-1,i}\right) \tag{2.5}$$

La fonction de coût E à minimiser dans le cas d'un apprentissage est une mesure de l'erreur entre la sortie souhaitée pour une forme et la sortie calculée par le réseau. L'erreur sur une forme p se quantifie généralement par une erreur quadratique $E_p(w)$:

$$E_p(w) = \frac{1}{2}\sum_{q=1}^{N_L}(o_{L,q}(x_p) - d_q(x_p))^2 \tag{2.6}$$

L'erreur pour l'ensemble des formes $E_p(w)$ est donc :

$$E(w) = \sum_{p=1}^{P} E_p(w) \tag{2.7}$$

Le problème se résume donc à :

$$\min E(w) \tag{2.8}$$

Pour résoudre ce type de problème, une technique classique d'optimisation issue de la recherche opérationnelle consiste à déterminer par itérations successives les valeurs du paramètre w. Au regard des objets à manipuler, la descente de gradient est une réponse adéquate à ce problème. Elle consiste à utiliser un point existant w_0 et lui faire effectuer un déplacement dans la direction de l'antigradient. Le nouveau point obtenu par la translation $w \to w - \mu\nabla E(w)$ a une plus petite valeur pour la fonction objectif. Le paramètre μ est un pas positif appelé dans le cas présent pas d'apprentissage. L'opération de translation est répétée jusqu'à l'obtention d'une solution satisfaisante.

```
Initialisation aléatoire des poids du réseau
répéter
    pour chaque échantillon de la base d'apprentissage faire
        Propager l'échantillon dans le réseau
        Calcul de l'erreur sur la couche de sortie
        Propagation de l'erreur sur les couches inférieures
        Ajustement des poids
    fin
    Mise à jour de l'erreur totale
jusqu'à Critère d'arrêt
```

ALGORITHME 1 – Apprentissage d'un PMC par rétropropagation du gradient

En utilisant la rétropropagation du gradient de l'erreur (Annexe C.1, p. 143), le résumé du déroulement de la méthode est donné par Algo. 1.

Bien que l'erreur soit minimisée localement, la technique permet de converger vers un minimum et donne de bons résultats pratiques. Dans la plupart des cas, peu de problèmes dus aux minima locaux sont rencontrés. Il persiste cependant deux problèmes que l'on rencontre dans une application réelle qui sont d'une part la lenteur de la convergence si μ est mal choisi et d'autre part le possible risque de converger vers un minimum local et non global de la surface d'erreur.

La fonction erreur quadratique ne possède qu'un minimum (la surface est un paraboloïde). L'algorithme est assuré de converger, même si l'échantillon d'entrée n'est pas linéairement séparable, vers un minimum de la fonction erreur pour un μ bien choisi. Si μ est trop grand, on risque d'osciller autour du minimum. La figure 2.10 illustre différents cas pouvant se produire avec une parabole pour différentes valeurs de pas d'apprentissage fixe.

Pour éviter des comportements oscillatoires autour de la solution sans pour autant que la convergence soit lente, une modification classique du pas d'apprentissage consiste à diminuer graduellement sa valeur en fonction du nombre d'itérations (Fig. C.2).

Le principal défaut de cette méthode est un temps de convergence restant assez long qui dépend de différents paramètres comme l'initialisation à l'instant $t = 0$ des poids synaptiques ou de la valeur initiale du paramètre μ. Il n'en reste pas moins qu'elle donne de bons résultats expérimentaux.

Dans une implémentation de l'algorithme de rétropropagation de l'erreur, il est aussi difficile de déterminer quand l'ajustement des poids du PMC doit s'achever. Plusieurs critères d'arrêt sont employés : les itérations cessent quand la norme du gradient est proche de zéro (les poids ne varient alors que très peu), ou bien alors dès que l'erreur en sortie est en dessous d'un certain seuil. Le premier critère est plus intéressant mathématiquement car il correspond à la stabilisation de la solution dans un minimum, le second est plus proche de critères réels (interprétables) de bonne corrélation entre solution calculée et solution attendue. Dans ce dernier cas, si le problème étudié concerne une tâche de classification, on peut considérer que l'apprentissage s'achève quand toutes les formes sont classifiées, ce qui permet de s'affranchir de la détermination du taux d'erreur à ne pas dépasser.

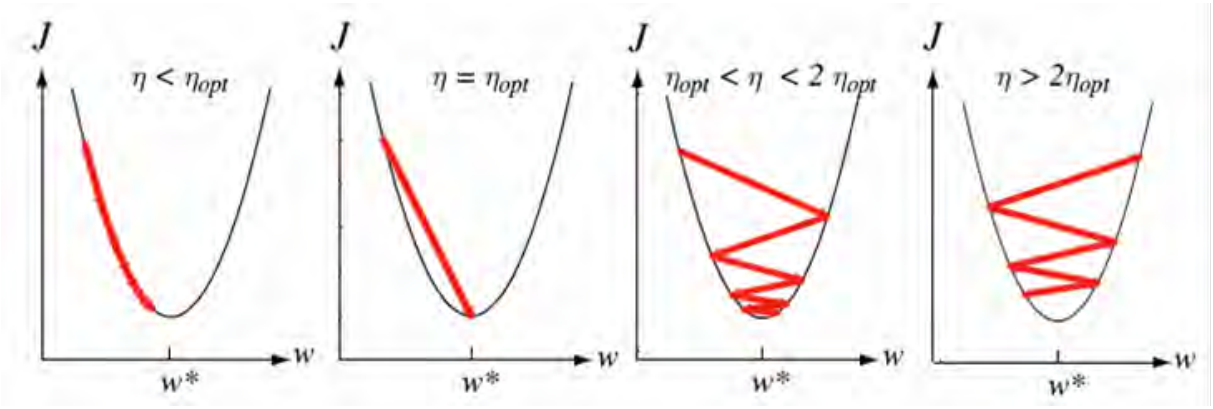


FIGURE 2.10 – Exemple de comportement en deux dimensions pour différentes valeurs de pas d'apprentissage

En pratique, on allie ce dernier critère d'arrêt à un deuxième qui tient compte d'un nombre maximum d'itérations à ne pas franchir. En effet, il n'est pas garanti que le réseau puisse classifier toutes les formes, même avec un nombre infini d'itérations. La combinaison des deux conditions permet d'obtenir une solution correcte dans un temps raisonnable.

La validation croisée est une technique s'assurant principalement de la bonne généralisation du réseau, c'est-à-dire de son bon fonctionnement sur de nouveaux échantillons. Elle consiste à utiliser deux bases : l'une pour l'apprentissage et l'autre pour le test d'arrêt. La première, comme son nom l'indique, sert uniquement à l'algorithme de rétropropagation du gradient, la seconde permet de tester, à la fin de chaque itération, la qualité du réseau. Tant que l'erreur globale du réseau sur la base de test diminue, les itérations continuent. Dès que l'erreur augmente, l'apprentissage est stoppé même s'il aurait été possible de diminuer encore l'erreur sur la base d'apprentissage (Fig. 2.11). Cette solution, quand on dispose d'un grand nombre d'échantillons, permet d'éviter le phénomène de surapprentissage *(overfit)* sur les données d'apprentissage ayant comme conséquence une mauvaise généralisation [Tetko et coll., 1995] (Fig. C.3). Les résultats en termes de taux de lecture ou de taux d'erreur sont alors donnés pour une troisième base, indépendante des deux autres, nommée base de validation.

Reste le choix des échantillons lors de l'apprentissage qui est aussi un problème crucial ; il existe dans ce domaine peu de résultats théoriques concernant la création d'une « bonne » base d'apprentissage. Il est évident que dans un cas réel, afin d'avoir une bonne fiabilité et un grand pouvoir de généralisation, les exemples doivent être d'autant plus nombreux que le problème est complexe et sa topologie peu structurée. Dans certaines situations, comme le sera la nôtre, où les échantillons sont en nombre réduit, il est très fréquent d'utiliser plusieurs fois les mêmes échantillons pour permettre la diminution de l'erreur globale du réseau. Pour éviter des phénomènes de surapprentissage de certaines classes, il est recommandé de fournir au réseau un nombre d'exemples similaire pour chacune des classes et de les présenter de manière aléatoire lors de l'apprentissage.

FIGURE 2.11 – Arrêt par la méthode de validation croisée : les itérations stoppent dès que le taux d'erreur sur la base de test augmente même s'il peut encore diminuer pour la base d'apprentissage

2.4.4 Applications

La plupart des applications en analyse et reconnaissance de documents utilisant une approche neuronale sont fondées sur des Perceptrons multicouches, des HMM *(hidden Markov models* [Rabiner, 1989]*)* ou des cartes auto-organisatrices [Kohonen, 2001]. Au niveau de l'analyse de la structure logique, nous avons vu que les approches dirigées par le modèle et principalement celles à base de grammaires sont prépondérantes dans la littérature, celles dirigées par les données sont finalement assez peu nombreuses et l'utilisation des réseaux de neurones est extrêmement rare. Cette situation s'inverse totalement dans le cadre de l'analyse physique ; peu de chercheurs comme [Kopec et Chou, 1994 ; Tokuyasu et Chou, 2001] utilisent des grammaires ou encore [Spitz, 1991] avec un système à base de règles pour effectuer cette tâche. De manière générale, les solutions dirigées par les données semblent être privilégiées, et plus particulièrement les approches neuronales comme le PMC, dès que les données à traiter sont de bas niveau comme par exemple les pixels de l'image ou que l'application se rapproche du traitement du signal [Marinai et coll., 2005].

On retrouvera de nombreuses applications dans le prétraitement de l'image comme : la binarisation [Chi et Wong, 2001 ; Hamza et coll., 2005], la réduction du bruit, la restauration de texte [Stubberud et coll., 1995] ou la suppression de lignes [Martin et Bellissant, 1991] qui sont tout aussi performantes que des méthodes reconstruction par suivi des bords [Whichello et Yan, 1996], la correction de l'inclinaison [Rondel et Bure, 1995 ; Palaniappan et coll., 2000] ou de la squelettisation bien que dans ce dernier cas, les méthodes non supervisées soient privilégiées [Ahmed, 1995 ; Datta et coll., 2001 ; Singh et coll., 2000]. Leur capacité d'apprentissage et leur habilité à généraliser un comportement à partir d'observations sont bien plus performantes que la résolution au cas par cas de toutes les combinaisons pouvant apparaître dans des applications réelles. Ils s'adaptent facilement aux variations et sont en plus particulièrement robustes au bruit. Ils parviennent également à gérer des données contradictoires et ceci même durant la phase d'apprentissage.

Les approches neuronales trouvent aussi de nombreuses applications dans la segmentation physique de l'image que ce soit au niveau de la classification de pixel [Etemad et coll., 1997 ; Jain et Zhong, 1996], de la segmentation de régions [Strouthopoulos et Papamarkos, 1998] ou la segmentation de pages [Liebowitz Taylor et coll., 1992 ; Cesarini et coll., 2001]. Ce type de méthodes permet d'obtenir des segmentations encore plus fines que celles en bloc comme [Nagy et coll., 1992 ; Breuel, 2003] ou polygonales comme [Akindele et Belaïd, 1993].

La segmentation de caractères (par dissection, par reconnaissance ou holistique [Casey et Lecolinet, 1996]), trouve aussi des solutions neuronales pour l'identification des caractères collés [Wang et Jean, 1993 ; Lu et coll., 1998] ainsi que la localisation des points de coupure [Eastwood et coll., 1997 ; You et Kim, 2003].

Pour la reconnaissance de lettres ou de mots, des articles de synthèse comme [Le Cun et coll., 1998] ou [Liu et coll., 2003] donnent un aperçu des techniques et des résultats obtenus par la littérature. Il est à noter que bien que certains systèmes donnent d'excellents taux de reconnaissance (Tab. B.1, Annexe B, p. 139) sur des caractères « propres », il reste encore à résoudre un bon nombre de points pour pouvoir obtenir de tels scores sur des images dégradées ou sur l'écriture manuscrite. En dehors des méthodes basées sur la segmentation des lettres, les méthodes de reconnaissance de mots holistiques sont la plupart du temps accomplies par des HMM [Saon et Belaïd, 1997 ; Anigbogu et Belaïd, 1995 ; Choisy et Belaïd, 2002] mais le PMC peut aussi servir à renforcer par combinaison [Kim et coll., 2000 ; Xu et coll., 2003] ou alimenter le HMM [Zhou et coll., 2001].

Dans le cas des méthodes à base d'extraction de caractéristiques, l'extraction et la phase d'apprentissage peuvent être mises en relation comme dans [Gori et coll., 2003]. La façon de construire l'entrée du réseau est aussi un point important pour la reconnaissance. Dans [Amin et coll., 1996], l'entrée brute est un graphe dont les nœuds sont des caractéristiques extraites du squelette du caractère et les arcs les représentations des relations spatiales entre les caractéristiques. D'autres approches utilisent des réseaux de neurones récurrents afin d'encoder des graphes plutôt que de simples séquences [Diligenti et coll., 2001].

Pour améliorer les capacités d'un réseau, il est possible de combiner plusieurs systèmes : le réseau de neurones peut soit être l'outil de combinaison de plusieurs experts [Lee et Srihari, 1995], soit la base des experts [Strathy et Suen, 1995], voire encore les deux à la fois [Mui et coll., 1994]. Ce principe est intéressant à retenir : les approches classiques multiplient les données d'entrée dans le but de donner l'information nécessaire et suffisante à un classifieur pour l'aider à prendre sa décision. Les résultats obtenus grâce aux méthodes de combinaison [Rahman et Fairhurst, 1999] montrent qu'il n'est généralement pas souhaitable de se baser uniquement sur un seul prédicteur mais que la collaboration de plusieurs systèmes permet d'accroître les taux de reconnaissance et ceci même s'ils partagent les mêmes données d'entrée. D'autres approches modifient directement la topologie du réseau plutôt que de se focaliser sur les données comme dans [Cecotti et Belaïd, 2005] où la topologie du réseau s'adapte par déplacement des connexions en tenant compte de l'erreur géométrique de l'image. Nous exploiterons une partie de ce raisonnement pour notre système : il est préférable de considérer le système faillible et qu'il sera nécessaire soit de le corriger pour qu'il s'adapte aux données, soit de corriger les données d'entrée pour qu'elles s'adaptent au réseau.

2.5 Conclusion

Contrairement aux réseaux à représentation distribuée comme les Perceptrons multicouches, il semblerait que les réseaux à représentation locale soient plus aptes à modéliser des principes

cognitifs et qu'il soit plus aisé d'en interpréter le comportement lors de la reconnaissance. Le modèle de [McClelland et Rumelhart, 1981] en est une illustration et se trouve être l'inspiration d'un certain nombre de systèmes de lecture.

Tout au long de ce chapitre, nous avons montré, au travers d'exemples de reconnaissance de mots manuscrits, comment des réseaux connexionnistes pouvaient être utilisés pour modéliser des principes cognitifs. Plusieurs points nous incitent à préférer ce type d'architecture :

- décomposer la reconnaissance en plusieurs niveaux de traitement en distinguant les processus de perception de bas niveau et les processus cognitifs de haut niveau ;
- intégration de connaissances à l'intérieur du réseau et coopération des différents niveaux du réseau pour adapter la reconnaissance aux différentes variations que peut prendre la forme ;
- utilisation d'un retour de contexte, intégration de la phase de segmentation pendant le processus de reconnaissance et ajustement des zones d'observation à chaque cycle perceptif par les mécanismes de génération, validation et insertion d'hypothèses ;
- possibilité d'expliciter les poids du réseau, de pouvoir interpréter chaque neurone, y compris ceux entre les entrées et les sorties ;

Que ce soit dans le système Perceptro ou le réseau de neurones transparent, ce type de fonctionnement a l'avantage d'être rapide et ne nécessite pas de prétraitement lourd avant son utilisation. La méthode a été testée sur des montants littéraux de chèques et donne des résultats convaincants. Le fait de revenir par retour de contexte sur les données d'entrée permet, dans leur cas, de reconnaître un plus grand nombre de formes et de contrôler finement la segmentation implicite du mot en lettres. L'utilisation d'extracteurs spécifiques comme l'a proposé Snoussi Maddouri en utilisant des descripteurs de Fourier montre comment profiter des cycles perceptifs pour rajouter de l'information uniquement quand la forme est difficile à reconnaître.

Le parallèle avec notre problématique est plus qu'évident : la reconnaissance de structures logiques repose elle aussi sur une segmentation de l'image. La classe de document à traiter et les règles générales d'édition sont des connaissances qui nous servent de contexte. La nature hiérarchique de la structure logique que l'on peut décrire comme un arbre est analogue à l'effet de supériorité du mot mis en avant pour la reconnaissance du manuscrit ; les principes des modèles cognitifs de lecture peuvent se généraliser au niveau de la page. La proximité de notre problème comparé à ceux vus précédemment, nous laisse penser que l'utilisation d'un réseau similaire au système Perceptro serait une solution viable.

Au cours des prochains chapitres, nous allons montrer comment étendre les principes évoqués jusqu'à maintenant pour construire un réseau capable de reconnaître les structures logiques de documents. Nous discuterons des possibilités d'amélioration et reviendrons sur certains choix faits par Côté ou Snoussi Maddouri qui peuvent s'avérer limitatifs. Plus qu'une simple adaptation, nous proposerons dans un premier temps un apprentissage adapté à ce réseau, comme Bouriel avait entrepris de le faire, en se basant sur les résultats théoriques du Perceptron multicouche. Reposant sur des bases mathématiques solides, le PMC est l'un des réseaux de neurones artificiels les plus utilisés pour résoudre des problèmes d'approximation, de classification et de prédiction et il est largement employé dans le domaine de la reconnaissance des formes et d'analyse de la structure physique.

Dans le système que nous voulons développer, nous considérerons les outils d'extraction comme faillibles : nous ne chercherons pas à obtenir les meilleures caractéristiques pouvant représenter le document avec des scores de confiance parfaits. Au contraire, nous nous limiterons à une analyse basique du physique avec des outils modestes afin de mettre plus en avant la méthode d'analyse du logique que la qualité de l'extraction physique. Nous tiendrons compte

dans notre système de l'imperfection de ses entrées et nous proposerons un système capable de remettre en cause ses résultats et de revenir à une nouvelle analyse de l'image afin de proposer de nouvelles entrées corrigées.

La principale critique que nous avons formulée à l'encontre de l'approche de [Côté, 1997] était la détermination manuelle et empirique des poids dans leur réseau et le non-recours à des liens inhibiteurs. La présence d'un apprentissage aurait pu pallier ces deux limitations. En contrepartie, c'est cet apprentissage fastidieux qui met en cause l'intérêt de l'emploi d'un PMC pour effectuer une analyse de la structure logique. Le système que nous devons proposer se doit donc de garder à la fois le côté perceptif du système de Côté tout en conservant la flexibilité et l'approche *data-driven* que peut proposer un PMC sans perdre en interprétabilité et en temps d'exécution. Nous allons montrer au prochain chapitre quelle architecture a été retenue pour proposer une méthode mixte entre le système Perceptro et un Perceptron multicouche.

Nous mettrons aussi en œuvre dans le chapitre 4 une méthode de partitionnement de l'espace d'entrée afin d'accélérer le processus de reconnaissance et de permettre une sélection des informations elle aussi perceptive. Nous proposerons aussi une topologie dynamique au chapitre 5, qui emprunte les concepts d'un réseau de neurones à décalage temporel, dans le but d'améliorer la prise en compte du contexte à l'intérieur même du réseau pendant la reconnaissance et l'apprentissage.

Chapitre 3

Réseau de neurones perceptif

Nous allons montrer au cours de ce chapitre comment adapter et intégrer les résultats sur la perception et la reconnaissance de l'écriture à notre problème d'analyse de structures logiques de documents. Sur les fondements des travaux de [McClelland et Rumelhart, 1981], [Côté, 1997] et [Snoussi Maddouri, 2003], nous élaborerons une nouvelle topologie neuronale, à mi-chemin entre la représentation locale et distribuée, qui permettra de s'adapter aux spécificités des formes que nous manipulons. Plusieurs concepts seront retenus comme l'utilisation du contexte, la décomposition de l'analyse en plusieurs couches, la gestion de l'ambiguïté. À ces principes, nous ajouterons un apprentissage du modèle et une correction de la segmentation de l'image en blocs pour construire celui que nous appellerons réseau de neurones perceptif.

Sommaire

3.1 Système proposé 53
3.1.1 Choix des primitives 54
3.1.2 Structures logiques 56
3.1.3 Contexte 56
3.1.4 Apprentissage 59
3.2 Reconnaissance par cycles perceptifs 61
3.2.1 Propagation 61
3.2.2 Analyse 62
3.2.3 Correction 63
3.2.4 Cycles perceptifs 65
3.3 Expérimentations 67
3.4 Conclusion 71

3.1 Système proposé

Dans le chapitre 2, nous avons déjà examiné le système Perceptro ainsi que son successeur le réseau de neurones transparent, en détaillant principalement les points que nous allions conserver à savoir : la représentation locale et l'organisation hiérarchique du réseau en couches, l'effet du contexte sur les formes à reconnaître, les cycles perceptifs avec la génération et la validation d'hypothèses, et la segmentation implicite par rétroaction sur l'image d'entrée.

Dans le cas de l'analyse de structures logiques de documents, plusieurs points doivent être modifiés pour pouvoir tenir compte de la nouvelle tâche de reconnaissance comme le choix des primitives, la topologie du nouveau réseau, l'analyse contextuelle qui donnera l'équivalent, pour notre problème, de l'effet de supériorité du mot, et l'apprentissage que nous apportons en nous appuyant sur le cas du Perceptron multicouche (Sec. 2.4, p. 43). La reconnaissance par cycles perceptifs et la correction de la segmentation seront elles aussi modifiées suite au changement de topologie et de fonction d'activation que nous proposons.

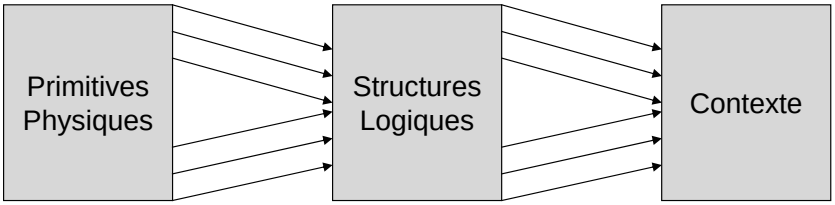


FIGURE 3.1 – Schéma général du réseau de neurones perceptif

3.1.1 Choix des primitives

Nous avons d'autres primitives, adaptées à l'analyse de structures logiques, qui proviennent essentiellement des résultats de l'analyse de la structure physique. Dans le but d'illustrer la méthode dans un cadre général, la majorité d'entre elles nous sont fournies par un OCR. Comme évoqué en sous-section 2.4.4 (p. 49), les méthodes permettant l'analyse de la structure physique et du texte ne manquent pas ; il existe pour ainsi dire une ou plusieurs méthodes différentes pour chacune des caractéristiques que nous extrayons [Kasturi et coll., 2002].

Nous définissons aussi plusieurs familles de primitives qui sont classées par ordre chronologique d'extraction et qui respectent une vision descendante, allant du global vers le local. On retrouvera les primitives :

- géométriques ;
- morphologiques ;
- textuelles.

Les primitives géométriques sont celles données par l'analyse physique du document, une fois la segmentation en blocs achevée. On retrouvera comme caractéristiques d'une boîte englobante :

- sa position (x, y) dans l'image ;
- sa dimension (largeur, hauteur) ;
- la longueur de l'espace vide par rapport aux blocs voisins dans les quatre directions (espace_haut, espace_bas, espace_gauche, espace_droite) ;
- l'encadrement de la boîte.

Les indices morphologiques sont des caractéristiques plus fines, liées à la forme de la lettre ou de la ligne, on y retrouvera :

- le style de la fonte (gras, italique, souligné, barré, petites capitales, couleur) ;
- le nom (Arial, Times, Tahoma, etc.) et la taille de la police ainsi que le mode (indice, exposant) ;
- l'alignement du texte et l'espacement entre les lignes de base ;

– les retraits gauche ou droit du bloc, l'alinéa de la première ligne, l'indentation verticale entre les paragraphes ;
– le nombre de lignes dans le bloc.

Les primitives textuelles sont celles que l'on peut extraire après avoir reconnu le texte, elles sont moins généralistes que les précédentes et dépendent plus de la classe de document :

– la langue majoritaire (Français, Anglais) ;
– le ratio de numériques, de capitales et de signes de ponctuation ;
– le pourcentage de mots connus (mots étant dans un dictionnaire de la langue majoritaire) ;
– la présence d'une puce ou d'une énumération ;
– la présence d'un mot-clé *(abstract, introduction, keyword, table, figure, conclusion, references, appendix)*.

En codant toutes ces informations dans des variables réelles, nous en avons en tout 56 possibles en entrée du système. Le codage est assez simple car la plupart des valeurs renvoyées par l'OCR sont déjà des réels dans $[0, 1]$. Pour les indices ayant des valeurs dans une liste comme le nom de la police ou la langue du bloc, nous créons autant de variables que de choix possibles, 0 désignant l'absence et 1 la présence d'un élément. Nous avons décidé de pondérer certaines variables comme par exemple la présence d'une puce. Au lieu de fournir comme indication simplement sa présence ou son absence, elle aura une forte valeur si elle est trouvée en début de bloc et diminuera de façon exponentielle lorsqu'elle s'en éloigne. La présence de mot-clés est elle aussi pondérée, la valeur dépend du nombre d'occurrences k du mot dans le bloc : $\sum_{i=1}^{k} k^{-2}$, et 0 si $k = 0$, un bloc contenant deux fois le mot-clé aura donc plus de poids qu'un bloc n'en contenant qu'un. Des variables comme la taille de la police sont normalisées afin d'être comprises dans l'intervalle $[0, 1]$. L'ensemble complet des variables varie donc dans le même intervalle $[0, 1]$.

Type	Variables
Géométrique	position, dimension, espacement, encadrement
Morphologique	style fonte, nom police, alignement, retraits, nombre lignes
Textuel	langue, ratio type de lettre, pourcentage de mots connus, présence de mots-clés, de puce, d'énumération

TABLEAU 3.1 – Indices physiques fournis par l'OCR et servant d'entrée au système de reconnaissance

Les trois niveaux de variables que nous proposons correspondent à trois niveaux d'analyse de l'image, les deux premiers sont complètement génériques et standard, les valeurs nous sont données par l'OCR commercial *ABBYY FineReader Engine 7.0* qui lui aussi effectue séparément l'analyse physique *(AnalyzePage)* de la reconnaissance du texte *(RecognizeBlocks)*. Toutes les primitives qu'il utilise trouvent une correspondance dans le schéma XML ALTO [Belaïd et coll., 2007]. Les primitives dont nous nous servons ne sont pas dédiées spécifiquement à la tâche de reconnaissance et proviennent donc d'un OCR utilisant des techniques simples. Nous avons voulu, surtout lors des tests, mettre l'accent sur les possibilités du réseau et non pas sur la pertinence des outils d'extraction en délaissant le rôle du classifieur.

Nous avons estimé qu'il était préférable de s'appuyer sur un ensemble réduit et « basique » de variables pour montrer qu'en partant de caractéristiques standard et rapides à extraire, le système est capable de reconnaître convenablement les structures logiques. Il est bien sûr évident que l'ajout d'autres indices physiques plus élaborés ou l'amélioration de ceux présentés donnera de meilleurs résultats.

3.1.2 Structures logiques

C'est au moment de la création de la topologie du réseau et surtout de sa ou ses couches de contexte, que l'aspect dirigé par le modèle de l'approche prend son sens. L'intégration de la connaissance se fait dans le placement et le concept porté par chaque neurone. Pour respecter les principes de McClelland et Rumelhart, la décomposition doit simplement se faire du local vers le global ou dit autrement, du spécialisé vers le générique. Pour la reconnaissance de document, la tâche est presque aussi facile que dans le cas de l'écriture, il suffit de déplier l'arbre de la structure logique sur le réseau. Si une DTD *(Document Type Definition)* est disponible, il suffit d'utiliser la hiérarchie des éléments pour construire le réseau.

Les sorties que nous voulons reconnaître sont les éléments composant la structure logique. Étant spécifiques à la classe de document, c'est la seule partie qui est à changer dans le réseau si l'on suit le schéma général de la figure 3.1. Contrairement à ce que propose la littérature, nous séparons les éléments jusqu'à la limite de la microstructure en proposant 21 classes permettant de couvrir tous les cas pouvant apparaître dans notre base de documents d'articles scientifiques (Annexe A). Le tableau 3.2 énumère tous les éléments de structure logique.

Titre du document	Auteur	Email	Adresse	Résumé
Mots-Clés	Catégories	Introduction	Paragraphe	Section
Sous-section	Sous-sous-section	Liste	Énumération	Flottant
Conclusion	Bibliographie	Algorithme	Copyright	Remerciements
Numéro page				

TABLEAU 3.2 – Éléments de structure logique choisis pour la base d'articles scientifiques

3.1.3 Contexte

Dans le cadre où nous nous plaçons, l'information pertinente pour classifier un objet (que ce soit un caractère, un mot, un paragraphe ou une illustration) est bien souvent extérieure à l'objet lui-même. Cette information réside parfois dans d'autres formes qui doivent elles-mêmes être classifiées mais elle peut également faire partie de l'environnement dans lequel travaille le classifieur.

Dans le premier cas, on peut améliorer la précision de la reconnaissance en prenant en compte les caractéristiques d'un groupe entier d'objets avant de classifier séparément chacun d'entre eux comme la classification par champs et la consistance de style proposées par [Sarkar et Nagy, 2005]. Dans le second cas, l'amélioration peut se faire en fournissant de la connaissance afin de spécialiser ou de raffiner le classifieur pour que la forme ou le groupe de formes correspondent mieux à leur environnement. Cette information supplémentaire est généralement appelée « contexte » et ceci même dans les situations où elle proviendrait des échantillons disponibles.

Si l'on reprend le cas de Côté, le contexte représente le mot. L'information du mot sur la lettre permet de mieux segmenter l'image de part la génération d'hypothèses sur la position des lettres. Ce contexte linguistique est très utile car, sans changer le classifieur, il permet de revenir à l'origine de l'information, de la réévaluer et finalement de reprendre une décision plus aisée. D'autres contextes sont possibles comme le modèle morphologique de [Katz, 1987], le modèle lexical, particulièrement utilisé en post-traitement des OCR [Rice et coll., 1999], ou bien encore le modèle syntaxique [Nagy, 1992].

L'information apportée par le contexte dans le cadre de la reconnaissance de structures logiques de documents est tout aussi importante et finalement assez simple à obtenir ou à reconstruire. La connaissance d'une structure générique peut servir de contexte. En effet, il est possible d'utiliser la nature hiérarchique de la structure logique pour trouver une information de contexte. Si l'on s'appuie par exemple sur un format éditorial comme la TEI, un document est composé d'un en-tête *(front)*, d'un corps *(body)* et d'une terminaison *(back)*. On dispose déjà d'un découpage très général en trois parties, qui sont à la fois communes à la structure physique et à la structure logique. Le corps d'un document peut lui-même être découpé récursivement de la même façon avec un en-tête qui englobe, pour un article scientifique, le titre, l'introduction et le résumé. Le corps contient le contenu du document et la terminaison correspond à la conclusion, la bibliographie et les annexes. Pour leur donner un nom simple et générique, nous les appellerons dans l'ordre : l'en-tête, la partie liminaire, le corps, l'appendice et la terminaison (Fig. 3.2). La signification de ces mots étant bien sûr différente en fonction de la classe de document traitée, c'est l'utilisateur qui fixe le sens de chaque partie de document, tout en respectant un principe d'emboîtement des neurones de la sortie par ceux du contexte.

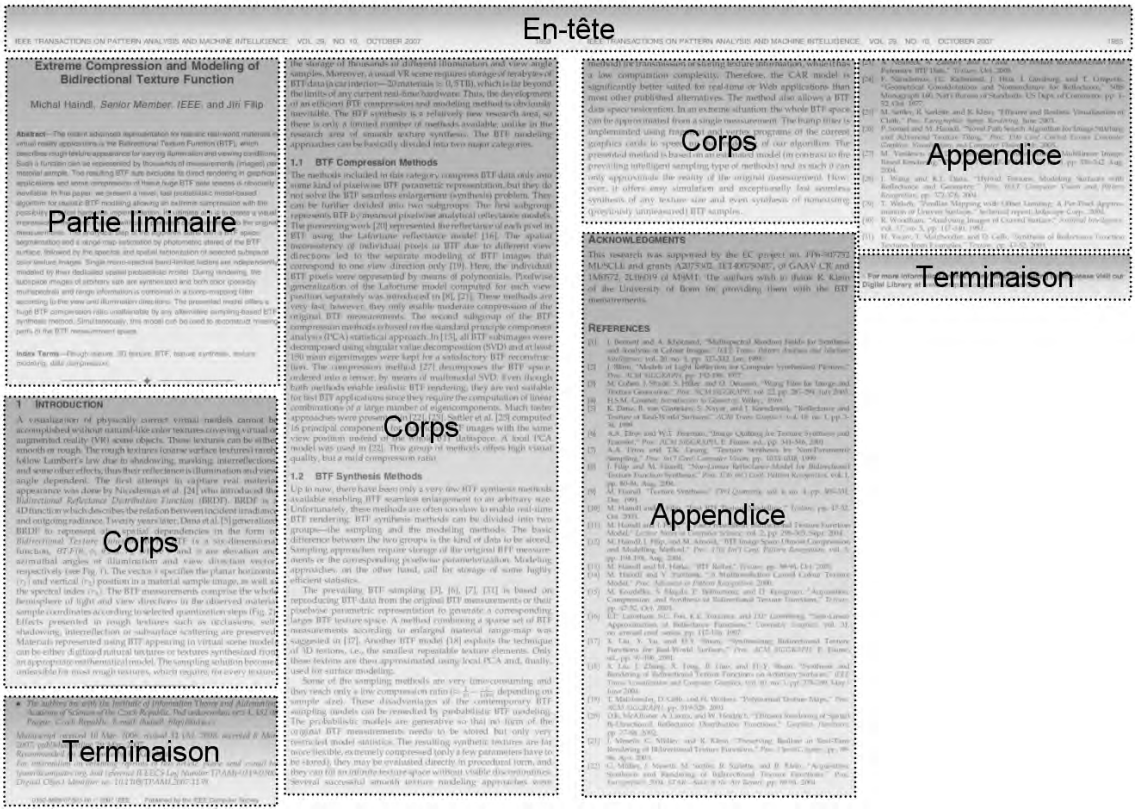


FIGURE 3.2 – Proposition de contexte pour un article scientifique

Nous conservons la topologie en couches d'abstraction avec une décomposition d'inclusion de gauche à droite : chaque élément de la couche n peut trouver un élément dans la couche $n+1$ plus générique qui l'inclut (Fig. 3.3). Pour le système Perceptro, trois couches sont utilisées comme dans le schéma que nous venons de présenter, nous avons vu que le réseau de neurones transparent en utilise quatre. Toujours en suivant les recommandations de la TEI, nous pouvons imaginer une couche supplémentaire entre les sorties (spécifiques au problème) et le contexte (général). On peut encore une fois découper chaque élément de la couche de contexte avec les cinq mêmes éléments. En fonction de la classe de document, il sera souvent difficile d'imaginer la signification de chacun des 25 éléments possibles, mais pour des documents volumineux comme une thèse, une annexe de fin de document peut être elle aussi être composée d'un en-tête, de son sommaire, d'un corps, de sa conclusion et de sa bibliographie.

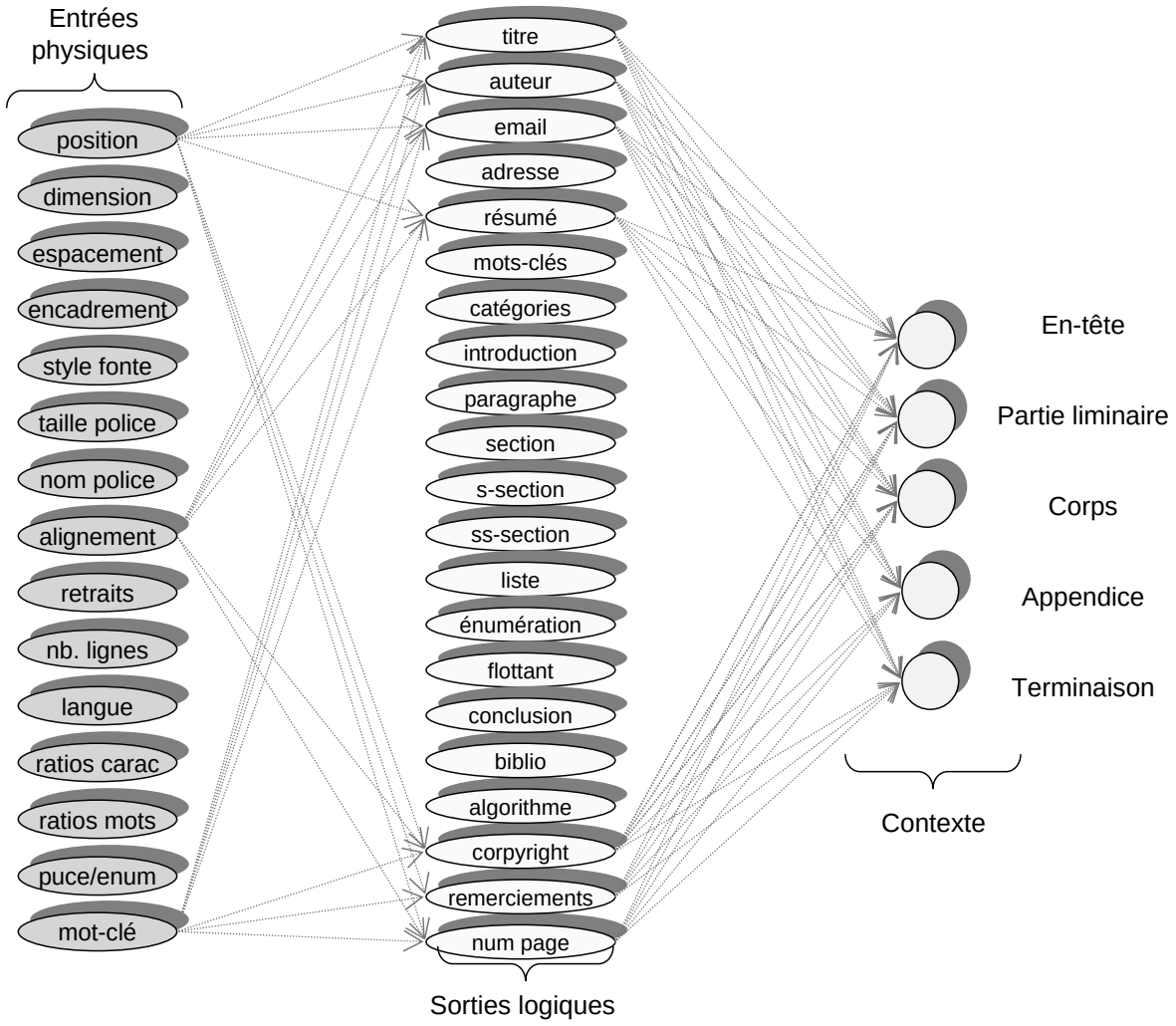


FIGURE 3.3 – Schéma spécifique du RNP pour l'analyse de structures logiques d'articles scientifiques

3.1.4 Apprentissage

Dans les réseaux des précédents auteurs, les liens sont fixés manuellement. L'utilisateur doit à la fois estimer s'il doit mettre un lien ou pas entre deux neurones ainsi que le poids de la connexion. De plus, les auteurs ne prennent pas en compte l'inhibition des connexions (Tab. 2.1, p. 40) alors que McClelland et Rumelhart la préconisaient. Pour orienter le réseau vers une approche dirigée par les données, nous avons décidé d'y ajouter un apprentissage afin de fixer les liens en rapport avec les valeurs réelles d'une base d'apprentissage. Comme le réseau est proche d'un Perceptron multicouche, nous allons utiliser les principes vus en sous-section 2.4.2 (p. 44) pour en modifier son comportement.

Dans ce système, que nous appellerons désormais réseau de neurones perceptif (RNP), nous remettons en place les connexions inhibitrices afin de stimuler la concurrence entre les neurones. La fonction d'activation n'est plus celle à saturation mais une fonction sigmoïde qui nous permet de profiter des propriétés du PMC (Sec. 2.4, p. 43). L'apprentissage se fait comme évoqué en section C.1 page 143, avec comme différence l'évaluation de l'erreur instantanée sur un poids :

$$\forall l, \forall p, \frac{\partial E_p(w)}{\partial o_{l,j}} = o_{l,j}(x_p) - d_j(x_p) \tag{3.1}$$

car pour chaque neurone du réseau nous connaissons la sortie attendue. Comme le calcul se fait localement à chaque neurone, la majeure partie des problèmes évoqués au chapitre 2 sont minimisés. Nous utiliserons aussi, à l'initialisation, un réseau totalement connecté. Nous laisserons l'apprentissage procéder éventuellement à la suppression des liens (Algo. 2, p. 75), plutôt que de prendre le risque d'oublier des connexions lors d'une construction manuelle comme l'ont fait les précédents auteurs.

L'avantage de cette solution, par rapport à un réseau à représentation distribuée, réside aussi dans le fait qu'il nécessite beaucoup moins d'échantillons pour son apprentissage, ce qui est très intéressant car l'élaboration d'une base de documents de vérité, pour l'apprentissage et le test, peut vite devenir assez fastidieuse. Le RNP reste de plus tout à fait interprétable, la figure 3.4 montre, sur des réseaux plus réduits, comment l'apprentissage se comporte sur un problème de classification de couleurs. Les données sont des triplets de composantes primaires RVB munies d'une étiquette de sortie parmi les sept couleurs de l'arc-en-ciel, le contexte est composé de deux neurones : chaud ou froid. L'état du réseau est montré à trois instants différents : à l'initialisation, à la cinquième et à la vingtième époque. L'épaisseur du trait est proportionnelle à l'intensité de la connexion, une couleur verte pour une excitation, rouge pour une inhibition.

À titre de comparaison, la même tâche est effectuée à la figure 3.5 mais en utilisant un Perceptron multicouche ayant le même nombre de neurones sur la couche cachée qu'en avait le RNP sur sa deuxième couche. On remarquera qu'il est très difficile d'interpréter les liens même sur un problème simple et une architecture réduite. On se souviendra aussi de la remarque faite sur la proposition de Bouriel et coll. (S.-Sec. 2.3.2, p. 42) à savoir qu'il n'est pas possible d'assigner tels quels les résultats d'un apprentissage d'un PMC sur un RNP même s'ils ont la même topologie et utilisent la même base de données.

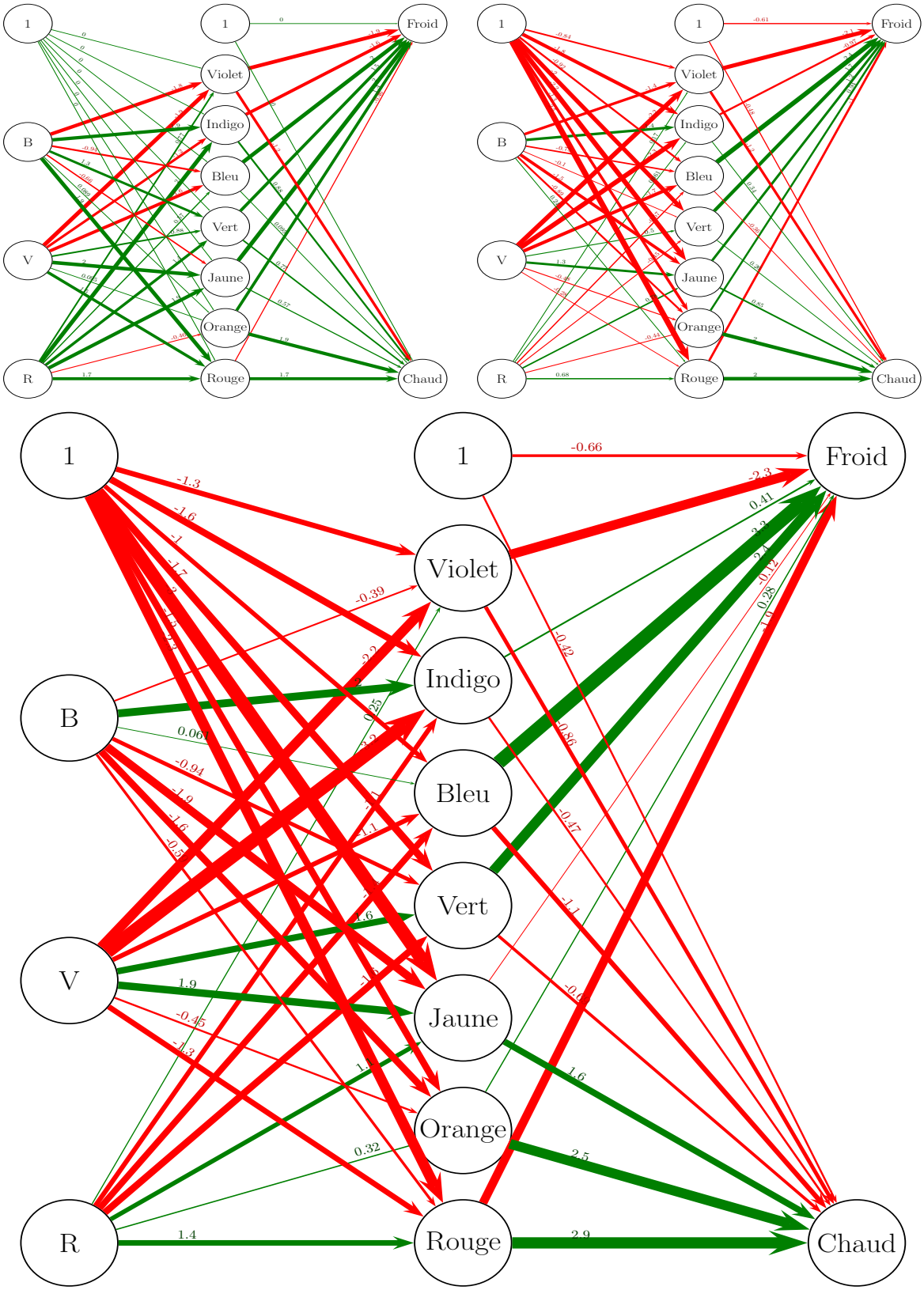


FIGURE 3.4 – Classification de couleurs par un réseau de neurones perceptif

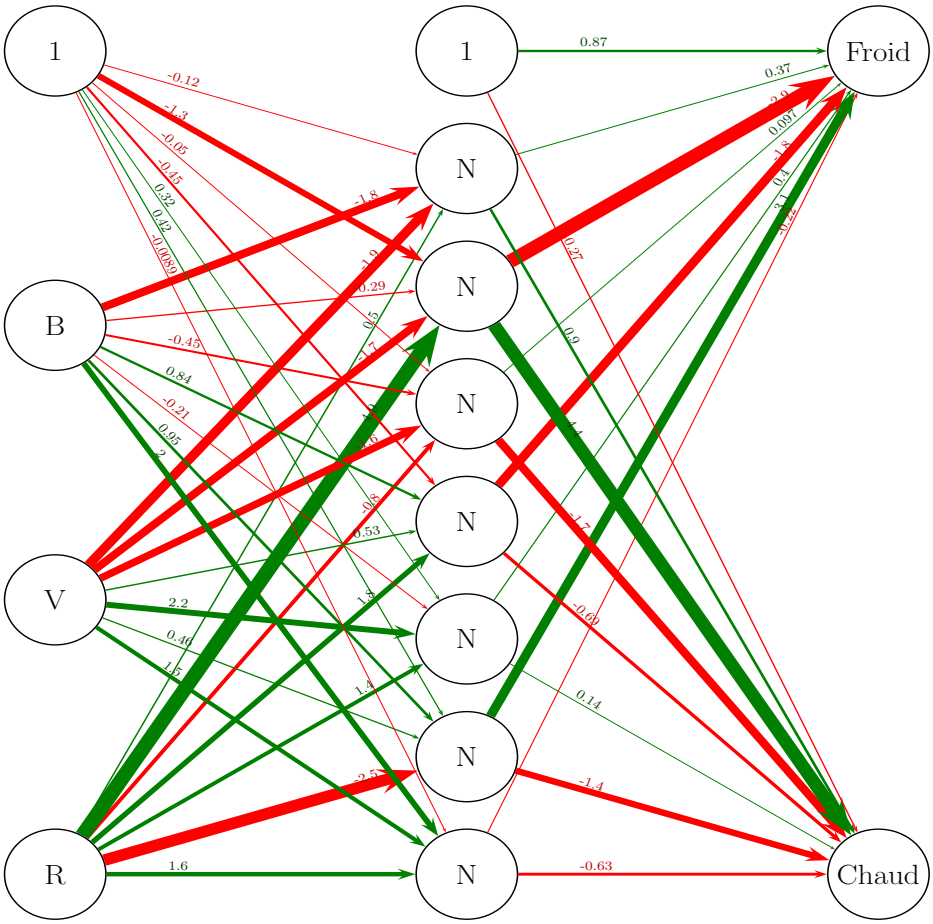


FIGURE 3.5 – Classification de couleurs par un Perceptron multicouche

3.2 Reconnaissance par cycles perceptifs

Une fois la topologie fixée et l'apprentissage terminé, le RNP est dans un état proche du système Perceptro. La reconnaissance sera similaire à ce dernier ; les informations seront propagées, les sorties analysées et si une ambiguïté est détectée, une génération d'hypothèses sera effectuée en fonction de l'information de la couche de contexte et de la couche de sortie dans le but de corriger l'entrée (Fig. 3.6).

3.2.1 Propagation

La propagation consiste à effectuer dans un premier temps une analyse physique de l'image du document : l'image est tout d'abord segmentée en blocs rectangulaires par l'OCR, chaque bloc est ensuite reconnu c'est-à-dire que des informations physiques sont extraites ainsi que le texte pour créer le vecteur d'entrée composé de 56 variables telles que données par le tableau 3.1 (p. 55). Ce vecteur alimente les neurones d'entrée du RNP puis l'information est propagée sur les couches suivantes.

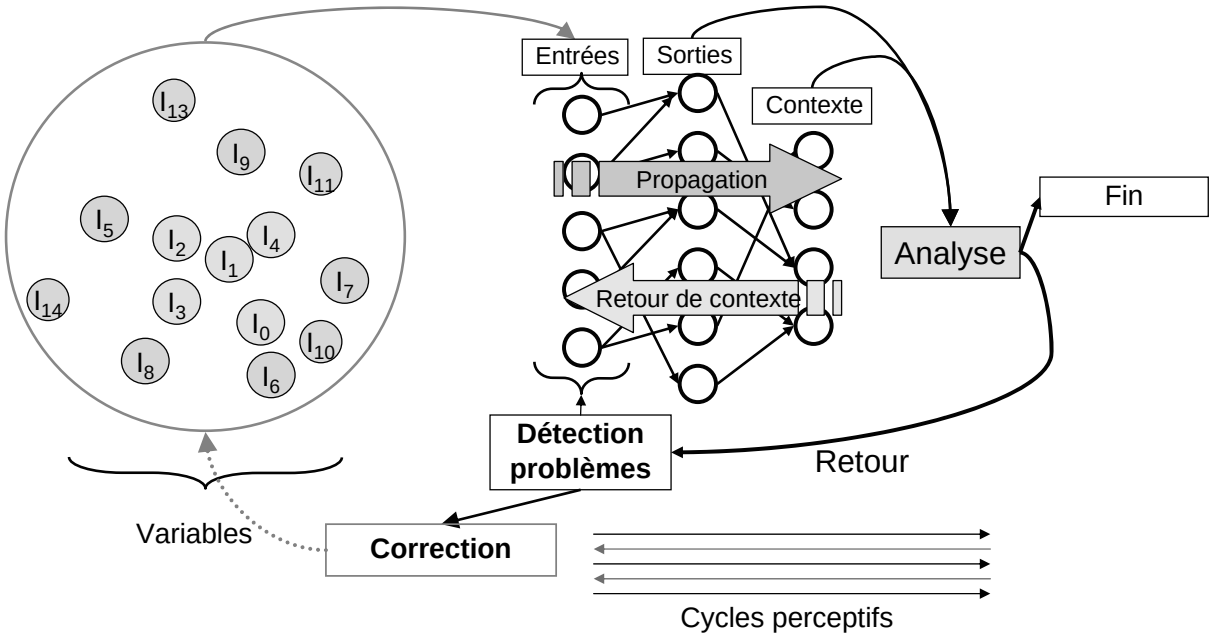


FIGURE 3.6 – Cycles perceptifs et RNP, extraction, propagation, analyse, correction

3.2.2 Analyse

Nous conservons les cycles perceptifs qui sont essentiels pour une bonne reconnaissance. Pour générer les hypothèses, nous utilisons une détection d'ambiguïté plus adaptée à la fonction d'activation du RNP, car nous ne pouvons plus compter sur la saturation des neurones à cause de la sigmoïde. Pour décider qu'une forme est correctement étiquetée, nous avons imposé deux conditions qui doivent être respectées.

Le premier critère M transpose à lui seul le fonctionnement du système Perceptro sur le RNP, la composante de la classe gagnante du vecteur de sortie O doit être la plus grande possible mais aussi supérieure à un certain seuil ε :

$$M(O) = \|O\|_\infty > \varepsilon \qquad 0 \ll \varepsilon < 1 \tag{3.2}$$

qui est une condition plus forte que celle consistant à n'utiliser que :

$$\text{argmax}_i\{O_i\} \tag{3.3}$$

ce qui permet de rejeter des formes donc l'étiquette supposée a un score de reconnaissance trop faible.

Le second critère Γ que nous imposons consiste à refuser un vecteur ayant sa composante gagnante trop proche d'autres composantes :

$$\Gamma(O) = \frac{n\left((\sum O_i)^2 - \sum O_i^2\right)}{(n-1)\left(\sum O_i\right)^2} < \eta \qquad 0 < \eta \ll 1 \tag{3.4}$$

comme le réseau ne donne pas des composantes de sortie très tranchées, η doit être relativement grand. Pour un vecteur de dimension 4, on a par exemple $\Gamma((0.9, 0.1, 0.1, 0.1)) = 0.55$ et $\Gamma((0.9, 0.8, 0.1, 0.1)) = 0.8$, de manière générale, si ε est suffisamment grand ($\varepsilon = 0.8$) nous avons remarqué expérimentalement que le critère Γ est très peu utilisé car il est rare pour une forme d'obtenir deux classes avec un score très élevé.

La conjonction des deux critères force l'obtention d'un vecteur de sortie proche d'un vecteur de base canonique. Dans ce cas idéal, on obtient $M(O) = 1$ et $\Gamma(O) = 0$ et le vecteur est accepté. Inversement, si les O_i sont tous identiques, $M(O)$ dépassera peut-être le seuil ε mais alors $\Gamma(O) = 1$, le vecteur sera refusé et la forme désignée ambiguë.

3.2.3 Correction

Quand un échantillon est rejeté par les critères M ou Γ, le vecteur de sortie est examiné et plusieurs classes potentiellement gagnantes sont retenues. Le contexte est ensuite utilisé : en fonction des neurones gagnants dans la couche de contexte, on élimine ou on conforte les classes retenues lors de la génération d'hypothèses, on se fixe alors la classe la plus probable. Pour corriger l'entrée, nous avons décidé de nous focaliser sur la segmentation des blocs car elle est la source principale des erreurs et des confusions les plus flagrantes [Kanai et coll., 1995 ; Yanikoglu et Vincent, 1998].

La segmentation est corrigée en fonction de l'hypothèse faite sur l'étiquette de la forme. Sachant la classe, la boîte englobante de la forme est modifiée en fonction de celle qui est représentative pour la classe considérée. Si la boîte est plus grande, elle sera divisée en deux parties, si elle est trop petite, elle sera agrandie.

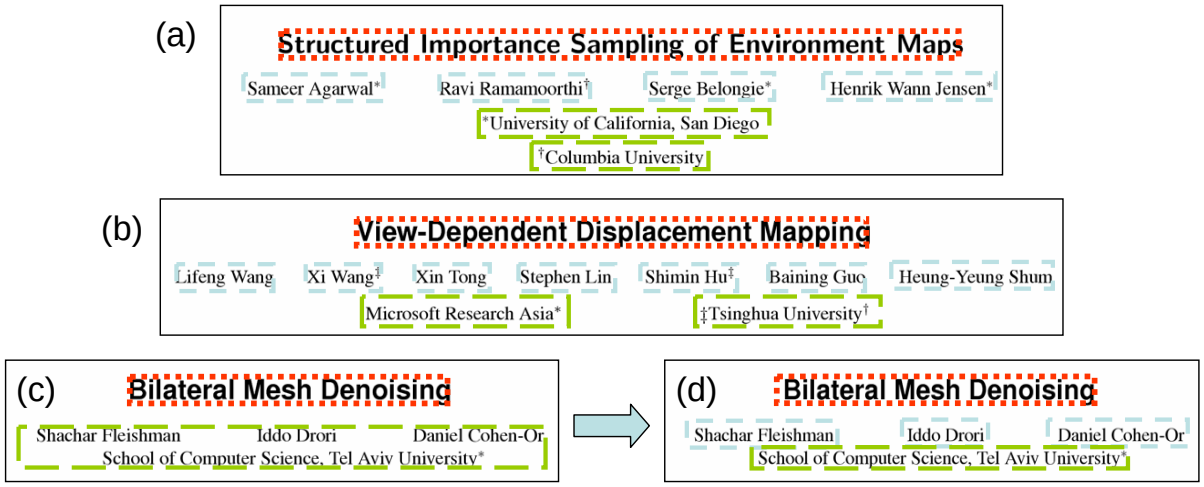


FIGURE 3.7 – Correction de boîtes englobantes. Dans les exemples (a) et (b), la segmentation et l'étiquetage sont corrects. Dans l'image (c), la boîte englobante du bas est trop grande, elle est divisée en deux parties et le processus est relancé sur les nouvelles segmentations. L'image (d) montre le résultat de la correction, le titre, les auteurs et l'adresse sont détectés et labellisés

Pour choisir les échantillons types, permettant de faciliter la correction de la segmentation, plusieurs techniques ont été expérimentées. Partir du réseau appris et tenter de retrouver le

meilleur échantillon «mathématiquement» ne donne pas de résultats exploitables dans le sens où ils ne correspondent pas à un exemple réel. Nous avons par exemple essayé une approche par algorithme génétique permettant de déterminer un échantillon parfait (tel que donné au réseau, la sortie sera un vecteur de base canonique). Le vecteur trouvé est en effet représentatif de la classe pour le réseau, mais il ne l'est plus par rapport à ce qu'il est censé représenter, ses valeurs sont totalement différentes de l'idée que l'on peut s'en faire. Même pour des données bas niveau comme les chiffres de la base MNIST, on peut créer de «faux» chiffres parfaits (Fig. B.3, p. 140).

Il se trouve que des techniques plus simples, comme prendre la moyenne de tous les échantillons d'une classe, donnent des résultats plus réalistes et surtout permettent une correction efficace. Dans les tests menés, nous avons choisi le médian comme représentant. Nous avons aussi pensé à utiliser plusieurs échantillons types par classe, afin de proposer plusieurs alternatives pour corriger. Un algorithme non supervisé comme le *k-means*, avec deux ou trois centres, pourrait être une bonne alternative à l'unique choix de l'échantillon médian mais qui demanderait plus de cycles perceptifs en cas de mauvais choix d'hypothèses de correction. Une autre possibilité serait de sélectionner les échantillons pendant l'apprentissage comme évoqué dans [Vajda et coll., 2006].

La modification de la boîte englobante peut se faire soit en agrandissant soit en réduisant sa taille. Pour le premier cas, nous n'avons pas rencontré de problèmes où la boîte donnée par l'OCR était plus petite que celle attendue. Si la situation se présentait, avec l'information que l'échantillon type aurait une boîte englobante beaucoup plus grande que l'actuelle, on augmenterait alors ses dimensions et on fusionnerait éventuellement le bloc courant avec des blocs voisins.

C'est le deuxième cas qui se produit classiquement avec l'OCR que nous utilisons : le bloc est beaucoup plus grand (sur-segmentation) et contient au moins deux structures logiques différentes (Fig. 3.8). Il faut donc diviser le divisier en s'aidant des échantillons types. En fonction du nombre de lignes contenue dans le bloc et la hauteur de l'échantillon type, on subdivise en deux parties le bloc trop gros.

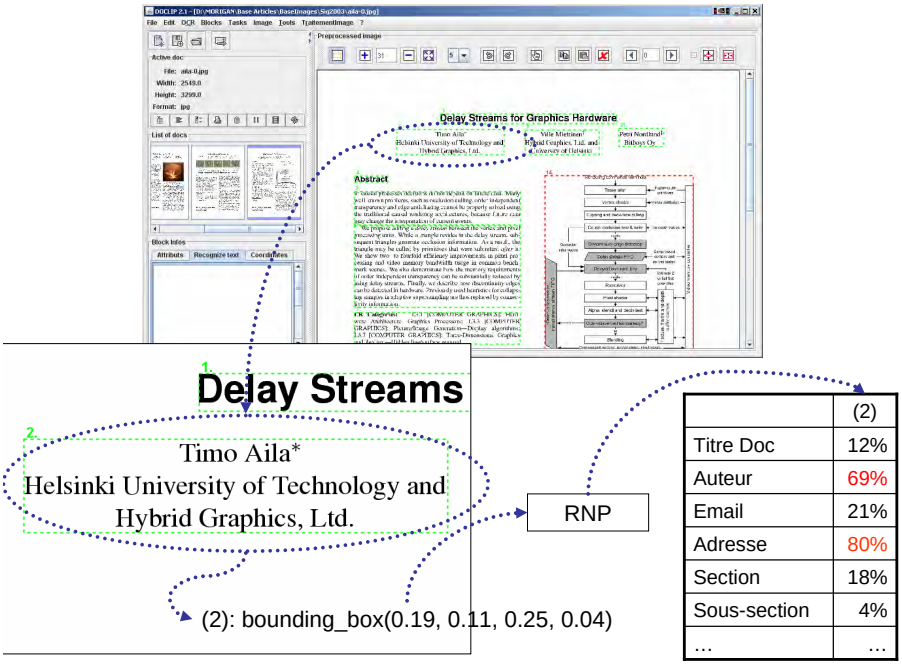


FIGURE 3.8 – Ambiguïté sur un bloc mal segmenté

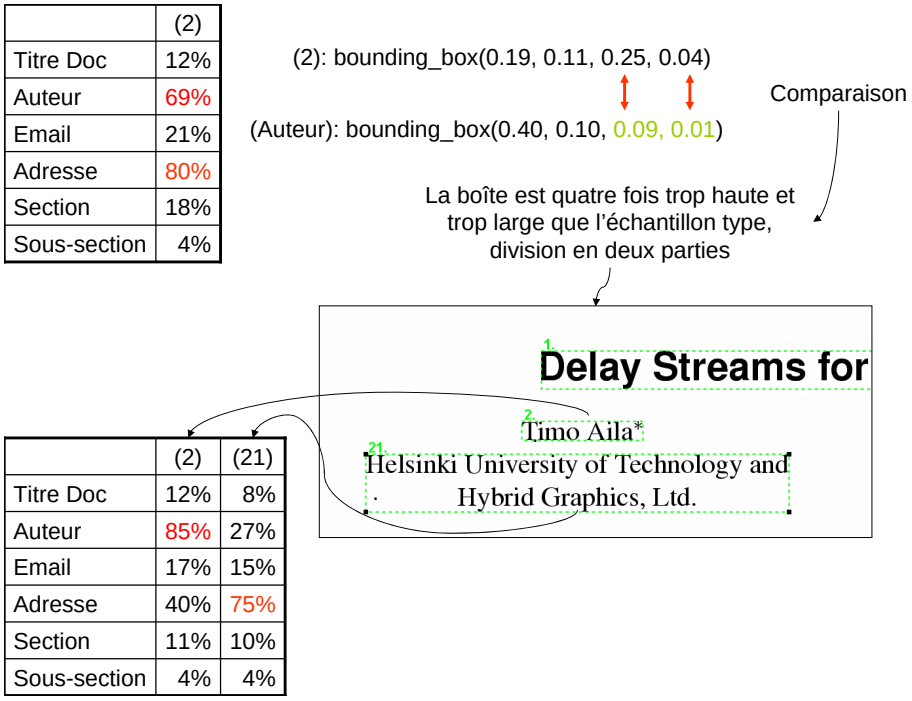


FIGURE 3.9 – Choix d'une hypothèse et correction de la segmentation par utilisation de l'échantillon type

Le premier découpage se ferra sur l'axe des ordonnées, de telle sorte que la nouvelle segmentation corresponde mieux à la hauteur de l'échantillon type (Fig. 3.9). Le RNP effectue ensuite la reconnaissance des deux blocs séparément et valide l'hypothèse. Si besoin est, un découpage horizontal peut aussi être effectué, même si dans notre cas il n'est nécessaire que pour séparer des noms auteurs trop proches sur une même ligne. Aucun autre problème de segmentation horizontale n'a été détecté bien que les documents soient sur deux colonnes.

Si une ambiguïté persiste entre deux classes, malgré les tentatives de correction, le contexte servira une dernière fois pour choisir entre deux classes qui seraient trop proches. Ainsi, si le système hésite entre une introduction et un paragraphe quelconque et si le contexte indique plus clairement que le bloc se trouve dans la partie liminaire, le réseau décidera de faire gagner la classe de l'introduction. La figure 3.10 montre comment le contexte joue en la faveur d'une section plutôt que d'une sous-section. Cette utilisation du contexte sert aussi à la validation d'hypothèses, l'activation des neurones de contexte et les poids des liens permettent d'utiliser un classement différent des hypothèses en choisissant en priorité celles qui correspondent le mieux avec ce qu'indique le contexte.

3.2.4 Cycles perceptifs

Si l'on occulte les principales différences entre le système Perceptro et le RNP (Tab. 3.3), le principe de la correction reste similaire ; plusieurs couples de processus ascendants et descendants (extraction–propagation–analyse–correction) sont effectués avec le réseau pour reconnaître les formes. Pour les plus simples d'entre elles, il se comportera comme un PMC, une seule propagation devrait suffire à la classification. Pour les formes plus complexes, plusieurs cycles seront

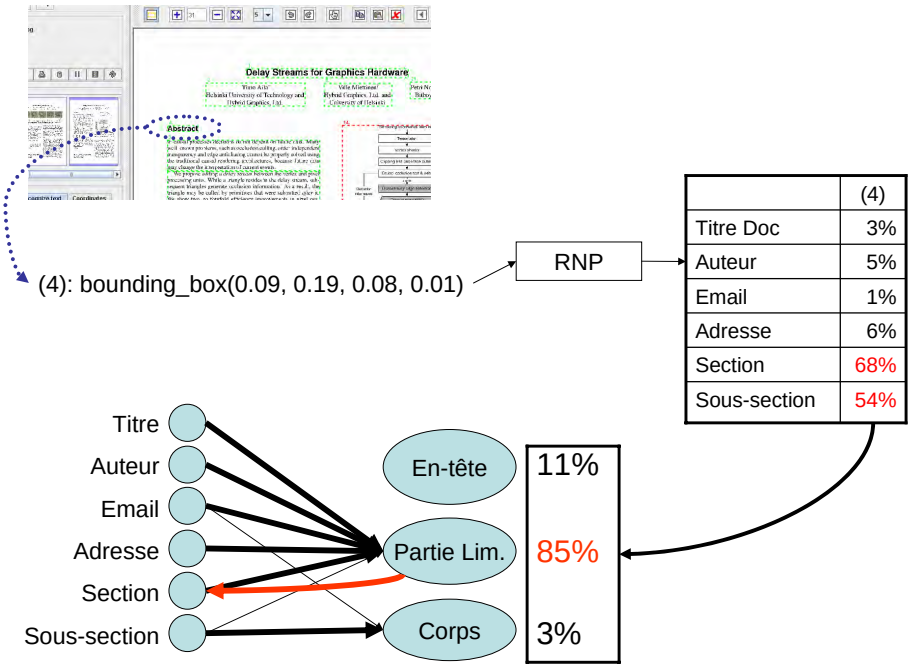


FIGURE 3.10 – Utilisation du contexte pour lever une ambiguïté

nécessaires. Chez les précédents auteurs, il fallait un nombre de cycles perceptifs assez élevé, de l'ordre d'une dizaine, pour aboutir à une solution. D'après nos expérimentations, il semblerait qu'il en faille trois fois moins pour obtenir des résultats corrects. Ceci s'explique en partie par le fait que nous corrigeons les entrées, en effectuant de nouvelles extractions des observations physiques. Dans le système Perceptro, les données extraites n'évoluent jamais, c'est la façon de les présenter au réseau qui change lors des cycles perceptifs.

Notre façon de procéder a parfois été assimilée à du *boosting* [Schapire, 1990 ; Freund, 1995]. Bien que le but soit aussi une amélioration de la reconnaissance, la manière de la mettre en œuvre n'est pas la même. Nous considérons aussi le classifieur comme imparfait et instable *(weak learner)*. Le *boosting* est une technique très employée dont la stratégie consiste à exécuter successivement l'algorithme d'apprentissage sur différentes distributions de probabilités des exemples d'apprentissage et de combiner les classifieurs obtenus en un seul modèle performant. L'algorithme le plus connu est *AdaBOOST*. D'autres techniques d'optimisation permettant d'améliorer ses performances par des méthodes de vote sont connues comme le *bagging (bootstrap aggregation)* [Breiman, 1996] ou l'*arcing (adaptive recombination of classifiers)* [Breiman, 1998]. Elles restent focalisées sur le classifieur et non pas sur les données d'entrée que nous tenons comme la source principale des erreurs.

Le RNP allie donc une approche par le modèle et par les données. L'apprentissage, lui, permet de mieux tenir compte des entrées sans pour autant perdre en interprétabilité. Les cycles perceptifs sont l'atout majeur du système : sans eux, le RNP serait moins performant qu'un PMC classique. Le fait de corriger les entrées en cas d'ambiguïté permet de gagner en précision. La prochaine section introduit les expérimentations effectuées sur une base de documents et présentera plus de résultats quantitatifs.

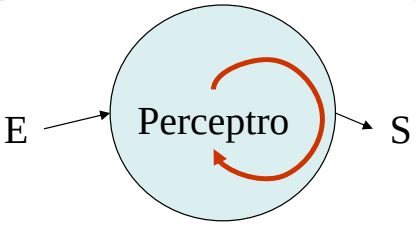
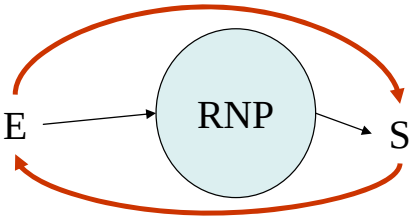
Perceptro	Réseau de neurones perceptif
Effet de la supériorité du mot	Hiérarchie de la structure logique
Pas d'apprentissage	Apprentissage
Saturation, pas d'inhibition, liens a priori	Sigmoïde, totalement connecté, inhibition, excitation
Cycles en interne	Cycles internes en support
Pas de correction des entrées	Correction des entrées
Reconnaissance dépend entièrement du réseau	Les entrées-sorties ont plus de valeur que le réseau
	

TABLEAU 3.3 – Résumé des différences majeures entre le système Perceptro et le réseau de neurones perceptif

3.3 Expérimentations

Pour effectuer les tests qui suivent, nous avons utilisé une base d'articles scientifiques provenant de la conférence Siggraph[8] qui dispose d'une mise en page et d'éléments logiques assez fournis. L'annexe A donne, pour illustration, les deux premières et dernières pages de trois articles. Ils ont été imprimés en 1 200 dpi puis numérisés à l'aide d'un copieur[9] en noir et blanc et à 600 dpi. Les structures que nous voulons reconnaître (Tab. 3.2, p. 56) ont été choisies de telle sorte que tous les blocs de la structure physique puissent avoir une étiquette, sans recouvrements. À partir de 74 articles, nous avons créé des documents de vérité et étiqueté plus de 3 000 blocs. L'apprentissage est effectué sur 44 documents, tandis que le test est mené sur les 30 autres restants. Pour donner des résultats avec le moins de biais possible, les résultats présentés correspondent à une moyenne des scores trouvés pour quatre couples différents de 40 et 30 documents. Les résultats pour un seul couple sont aussi une moyenne de différents *bootstraps* concernant les paramètres du réseau, comme l'initialisation aléatoire des poids.

Pour comparer les résultats, le réseau de neurones perceptif est mis en concurrence avec un Perceptron multicouche standard, possédant deux couches cachées de 50 et 30 neurones. Le tableau 3.4 donne un résumé des résultats obtenus pour les deux systèmes [Rangoni et Belaïd, 2005 ; Rangoni et Belaïd, 2006].

Le PMC obtient de meilleurs résultats que le RNP sans cycle perceptif (au cycle n°1) ce qui s'explique simplement par le fait que ce premier dispose d'une topologie moins restrictive que celle du RNP. Les deux couches cachées permettent d'avoir une meilleure flexibilité dans la détermination des associations entre les observations physiques et les interprétations logiques. Lorsque l'on entreprend de faire plusieurs cycles et ainsi de dépasser le fonctionnement du simple PMC, le RNP gagne en taux de reconnaissance : dès le deuxième passage, le taux remonte grâce

[8] ACM SIGGRAPH, Special Interest Group on GRAPHics and Interactive Techniques, `http://www.siggraph.org/s2003/`

[9] Gestetner DSm745

Classes	PMC	Réseau de neurones perceptif			
		Cycle 1	Cycle 2	Cycle 3	Cycle 4
Toutes	81,7%	68,3%	79,2%	89,1%	90,8%
La meilleure	98,9%	85,3%	100,0%	100,0%	100,0%
La plus mauvaise	0,0%	0,0%	0,0%	28,9%	28,9%
Facteur temps	1	1	1,8	2,4	2,9

TABLEAU 3.4 – Classification de structures logiques par un PMC et un RNP avec cycles perceptifs

aux corrections effectuées à la fin du premier cycle. Au bout du troisième, on dépasse largement les résultats du PMC de près de 10%. Une classe jamais retrouvée (les emails) par le PMC se voit être reconnue par le RNP grâce aux corrections effectuées sur la segmentation.

Les classes les moins bien reconnues sont les emails, les adresses (Fig. 3.11) et toutes les structures logiques n'ayant pas un aspect physique permettant de les reconnaître comme la conclusion ou l'introduction (Tab. 3.5). Pour les deux premières classes, elles sont premièrement largement sous-représentées dans la base d'apprentissage (car elles n'apparaissent que très rarement dans les articles) et posent en plus des problèmes de segmentation (fusion). La conclusion est la pire des classes, elle est quasiment toujours confondue avec un simple paragraphe et ceci se retrouve aussi pour les blocs d'introduction et de remerciements. Il faudrait, à notre avis, interpréter le texte pour la détecter car des observations sur la forme ne sont pas suffisantes (il n'y a aucun problème de segmentation pour cette classe).

Les classes les mieux reconnues sont le titre du document, le résumé, les titres de premier niveau, le numéro de page et le copyright (Fig. 3.12). Les indices physiques sont suffisamment bien typés pour séparer les classes. Les échantillons de ces classes sont d'ailleurs presque tous reconnus dès le premier cycle (plus de 90%) et totalement au deuxième cycle. Après le quatrième cycle, les résultats n'évoluent plus, les blocs non reconnus sont définitivement rejetés. Si l'on prend le cas des adresses emails, celles non reconnues par le RNP (et le PMC), le sont par manque d'information ; il faudrait interpréter le texte pour différencier les deux éléments logiques (la détection du caractère '@' pourrait être une solution simple et efficace). Comme évoqué en début de chapitre, le choix des indices physiques est un facteur déterminant pour obtenir de meilleurs taux de reconnaissance. D'ailleurs, en testant sur la base d'apprentissage, le système n'arrive pas à obtenir des taux moyens de plus de 95%. Dans nos expérimentations, l'accent est mis sur l'amélioration apportée par le réseau perceptif sur un modèle classique de type PMC.

Les facteurs de temps sont donnés en considérant le temps mis par le PMC comme référence. Ils dépendent entièrement du temps mis par l'OCR pour extraire les indices physiques. Il faut également souligner que l'OCR se comporte comme une boîte noire et que nous disposons de très peu de paramètres pour le contrôler. Nous avons aussi constaté que le temps mis à reconnaître seulement quelques blocs de l'image est du même ordre que celui mis pour traiter l'ensemble de la page. Il est donc assez difficile de juger la pertinence des facteurs de temps. Ils sont donnés à titre indicatif et reflètent une situation défavorable dans laquelle les outils d'extraction ne sont pas indépendants et ne sont pas paramétrables par l'utilisateur. On notera que, même dans cette situation, les temps des cycles perceptifs restent raisonnables ; en doublant le temps de reconnaissance, le score de reconnaissance augmente significativement.

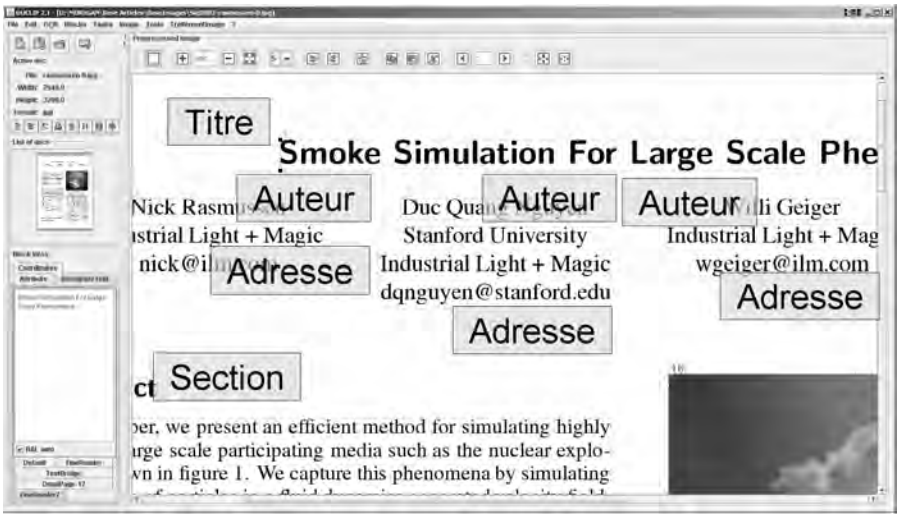


FIGURE 3.11 – Document comportant une erreur d'étiquetage. Les emails sont fusionnés avec les adresses plus hautes, ils portent donc la même étiquette

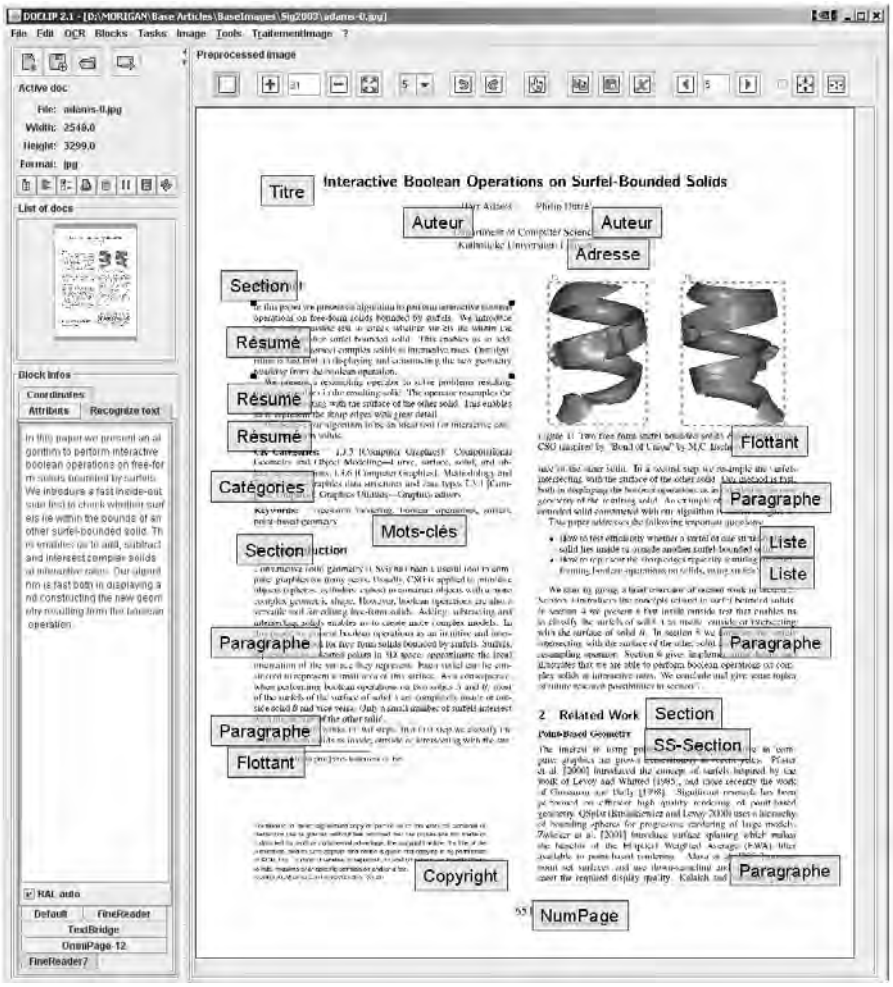


FIGURE 3.12 – Document parfaitement labellisé

Classes	Nb. échantillons	Taux PMC	Taux RNP
Titre Document	15	93,3%	100,0%
Auteur	44	88,6%	90,9%
Email	5	0,0%	80,0%
Adresse	21	47,6%	76,2%
Résumé	15	93,3%	100,0%
Mots-clés	14	92,8%	92,9%
Catégories	9	88,9%	100,0%
Introduction	73	80,8%	80,8%
Paragraphe	440	96,1%	97,1%
Section	92	97,8%	97,8%
Sous-Section	62	98,3%	98,4%
Sous-sous-section	17	76,4%	76,5%
Liste	69	97,1%	98,6%
Énumération	44	95,4%	97,7%
Flottant	105	91,4%	99,1%
Conclusion	38	28,9%	28,9%
Bibliographie	187	98,9%	98,9%
Algorithme	86	95,3%	94,2%
Copyright	9	88,8%	100,0%
Numéro page	30	96,6%	93,3%
Remerciements	10	70,0%	70,0%

TABLEAU 3.5 – Résultats détaillés du réseau de neurones perceptif au troisième cycle pour chaque classe

La recherche des indices physiques se fait à l'aide de *FineReader 7* dans sa version *Engine* qui consiste en un exécutable en ligne de commande dont l'entrée est l'image du document et renvoie ses résultats d'analyse physique et de reconnaissance du texte dans un fichier XML propriétaire. Les données sont complétées et normalisées par un logiciel *XMLExpand* que nous avons développé pour alimenter le réseau de neurones perceptif qui lui aussi a été développé indépendamment en C++. Une interface graphique en Java permet d'utiliser simplement l'ensemble de ces logiciels en lignes de commande. Elle permet aussi de créer manuellement les documents de vérité et de prendre en charge les conversions de format entre les différents modules. Pour rester dans des communications standardisées, l'échange des données entre les différents modules composant le système se fait à l'aide des formats précédemment cités : ALTO pour la structure physique et TEI pour la structure logique, les liens entre les deux structures étant maintenus par un fichier METS. Une vue d'ensemble de l'utilisation des trois formats est donnée par la figure 3.13, le passage d'un format propriétaire vers une version normalisée se faisant par l'intermédiaire de feuilles de transformation XSLT [Belaïd et coll., 2007].

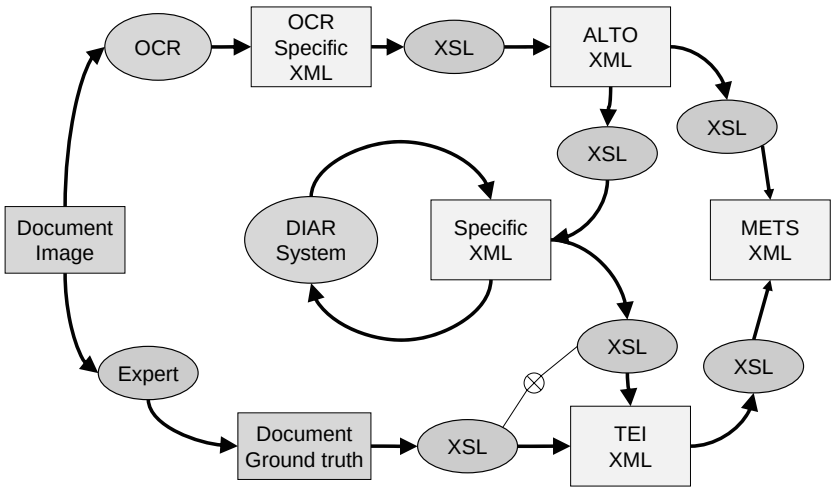


FIGURE 3.13 – Utilisation des normes ALTO, TEI et METS pour l'échange des données

3.4 Conclusion

À partir de travaux menés sur la perception humaine lors de la reconnaissance de l'écriture, nous avons utilisé le modèle d'activation interactive de [McClelland et Rumelhart, 1981] pour construire notre propre réseau de neurones. L'architecture et le fonctionnement de ce dernier sont similaires à ceux du système Perceptro de [Côté, 1997], repris par [Snoussi Maddouri, 2003] plus tard. Des changements nécessaires ont été effectués pour adapter l'existant au problème de reconnaissance de structures logiques de documents, notamment le choix de primitives ainsi que l'organisation et la sémantique des neurones dans le réseau. L'analyse contextuelle utilise désormais la nature hiérarchique de la structure logique pour organiser les couches composant le contexte. Le principe des cycles perceptifs a aussi été conservé et nous avons choisi de nous concentrer sur la correction de la segmentation qui s'avère être la source majeure des problèmes rencontrés.

Partant de données bruitées, nous n'avons ni gardé la fonction d'activation à saturation, ni la manière de fixer les poids des précédents auteurs ; nous avons proposé un apprentissage, proche de celui du Perceptron multicouche, permettant au réseau de déterminer lui-même les relations entre les observations physiques et les interprétations logiques. Le réseau modifié, que nous avons nommé réseau de neurones perceptif, est plus apte à traiter des données d'entrée imparfaites. Il conserve une architecture à représentation locale avec intégration de connaissances dans les neurones tout en ayant une forte prédominance *data-driven*. La détection d'ambiguïté et la correction de la segmentation ont aussi été revues pour être en adéquation avec le nouveau fonctionnement du réseau.

Les améliorations apportées au réseau de neurones perceptif font de lui une solution hybride entre une méthode *data-driven* et *model-driven* avec une architecture à mi-chemin entre représentation locale et représentation distribuée. Les résultats obtenus sur notre base de test confortent le fait que le choix d'une approche perceptive pour notre problème est tout aussi profitable que dans le cas de la reconnaissance du manuscrit. Si l'on compare les résultats obtenus à ceux d'expérimentations similaires présentées au chapitre 1 on s'approche de résultats comme ceux

de [Ishitani, 1999] ou de [Kim et coll., 2001] en notant toutefois que la majorité n'utilise pas autant de structures que nous (en moyenne 7 contre 21 chez nous) et qu'elles considèrent aussi que les données sont parfaites. Nous n'utilisons d'ailleurs que très peu d'informations concernant la microstructure contrairement à ce que la littérature propose, et nous avons insisté sur le fait que tous nos indices dépendent principalement d'un seul et même OCR commercial. Le but de la comparaison que nous avons faite avec le PMC est de montrer le gain potentiel de reconnaissance que le réseau de neurones perceptif peut apporter à une approche très peu employée dans la littérature.

Le gain de reconnaissance que nous obtenons se fait au détriment d'un allongement du temps de reconnaissance. La correction des entrées ou la validation d'hypothèses peut nécessiter plusieurs extractions des indices physiques. De plus, l'allongement peut en théorie être multiplié par le nombre de cycles perceptifs utilisé si tous les blocs, à chaque passage, ont besoin d'une nouvelle extraction. Nous estimons que pour notre base de documents scientifiques, le nombre d'extractions supplémentaires est multiplié par deux pour le troisième cycle perceptif pour lequel on obtient déjà de meilleurs résultats qu'avec un Perceptron multicouche. Nous allons montrer au prochain chapitre comment diminuer ce temps de reconnaissance en limitant les extractions physiques inutiles. L'idée développée consistera à créer une partition des entrées servant à alimenter progressivement le réseau par des groupes de variables, et à n'utiliser les extractions lourdes que si la forme est difficile à reconnaître.

Chapitre 4

Méthode de partitionnement

Au cours du précédent chapitre, nous avons exposé le fonctionnement du réseau de neurones perceptif en nous focalisant sur sa topologie, son apprentissage et sa reconnaissance singulière par correction des entrées et validation des sorties. Nous allons montrer dans ce chapitre comment les cycles perceptifs, qui sont l'atout majeur du réseau, peuvent être effectués plus rapidement et rendre le système encore plus proche de la vision humaine. Le partitionnement des variables d'entrée que nous proposons, issu d'approches de sélection et de réduction de données, permettra de conserver tous les concepts et les propriétés vues jusqu'à présent tout en réduisant la charge de travail au niveau de l'extraction des indices physiques.

Sommaire

4.1 Réseau de neurones perceptif et temps de reconnaissance . . . 73
4.2 Accélération de la reconnaissance . . . 74
4.3 Méthodes diminuant la taille de l'entrée . . . 76
4.3.1 La sélection de variables . . . 76
4.3.2 Classement de variables . . . 76
4.3.3 Sélection de sous-ensembles de variables . . . 78
4.3.4 Réduction de données . . . 80
4.4 Partitionnement de l'espace d'entrée . . . 83
4.4.1 Contraintes sur le choix de la méthode à proposer . . . 83
4.4.2 Justification de la méthode . . . 85
4.4.3 Algorithme de la méthode . . . 87
4.4.4 Choix de la dimension du sous-espace . . . 91
4.5 Expérimentations . . . 93
4.6 Conclusion . . . 100

4.1 Réseau de neurones perceptif et temps de reconnaissance

Il est de plus en plus fréquent que des travaux manipulant à la fois un grand nombre de données et de variables aient recours à des techniques de réduction de l'espace d'entrée. Les systèmes sont alors alimentés par des ensembles de taille beaucoup moins importante mais tout aussi informatifs et peuvent ensuite traiter avec plus de facilité le flot de données.

Les systèmes de reconnaissance dépendent généralement, en complexité, de la taille des entrées à traiter. Que ce soit au niveau de la complexité temporelle ou spatiale, ils sont rarement linéaires et toute réduction de l'espace d'entrée, même minime, peut entraîner des gains non négligeables en termes de temps ou de place mémoire. Ces gains sont d'autant plus appréciables pour des systèmes polynomiaux ou exponentiels.

Dans notre système de reconnaissance de structures logiques de documents, nous avons porté notre choix sur une solution à base de Perceptron multicouche. L'une des contraintes majeures de ce type de classifieur vient du fait que plus l'entrée d'un réseau est grande, plus le temps de reconnaissance et surtout d'apprentissage est long.

La complexité d'un PMC est polynomiale en son nombre de poids n. Elle dépend aussi du nombre de neurones k présents dans le réseau, ce qui au final donne une complexité moyenne en $\mathcal{O}(kn^3)$. Il faut aussi noter que le nombre de neurones k peut lui aussi être grand. En effet, comme évoqué en sous-section 2.4 p. 43, certains problèmes nécessitent un nombre exponentiel de neurones pour être résolus avec une seule couche cachée. Même s'il est toujours possible de transformer ce type de réseau sur plusieurs couches, le nombre de neurones sera quand même polynomial en son nombre d'entrées. Selon la nature du problème à résoudre, derrière la constante k se trouvant dans la complexité du PMC, se cache un nombre qui peut aussi croître fortement quand la taille de l'entrée et la complexité du problème augmentent. Ces considérations sont à envisager dans le pire des cas ; il n'en reste pas moins que la reconnaissance et surtout l'apprentissage d'un PMC requièrent un temps de calcul extrêmement long bien que chaque traitement au niveau du neurone soit élémentaire.

4.2 Accélération de la reconnaissance

Les seuls moyens permettant de réduire de manière significative ce temps polynomial sont soit la diminution de la taille de l'entrée (et indirectement le nombre de neurones dans les couches suivantes) soit la diminution du nombre de neurones dans les couches cachées ou bien encore les poids du réseau. Il n'existe pas de solution polynomiale pour déterminer le nombre optimal de neurones nécessaires dans la topologie (Sec. 2.4, p. 43). Réduire le nombre de poids demande des connaissances a priori sur les liens entre les neurones, ou bien alors de mettre en œuvre des techniques d'optimisation coûteuses. Pour cette dernière alternative, les solutions les plus connues sont l'*Optimal Brain Damage* de [Le Cun et coll., 1990b] et l'*Optimal Brain Surgeon* de [Hassibi et Stork, 1993] qui consistent toutes deux à supprimer, pendant l'apprentissage, les poids inutiles *(low-salency)* au réseau (Algo. 2).

Dans les travaux de [Côté, 1997] et [Snoussi Maddouri, 2003], les auteurs ont fait le choix de fixer eux-mêmes les liens ainsi que leur valeur avec comme justification la réduction de la complexité. En prenant le parti d'utiliser un apprentissage et de donner une prédominance aux données à notre système de reconnaissance, il nous a paru plus judicieux de ne pas intervenir sur les connexions du réseau et de laisser l'apprentissage se charger seul de la détermination des poids. La technique la plus fréquemment utilisée dans ce type de cas consistera à diminuer en prétraitement la taille de l'entrée.

Il est communément admis que le temps d'apprentissage est très lent, si bien qu'il n'est d'ailleurs pratiquement plus évoqué dans les travaux. Les auteurs préféreront se concentrer sur le temps de reconnaissance qui doit, lui, être le plus court possible. L'apprentissage est donc plus souvent considéré comme un prétraitement, et ce à juste titre, car il est fait une et une seule fois. Son résultat est ensuite exploité lors de la reconnaissance ce qui la rend à son tour

répéter
  Choisir une architecture conséquente pour le réseau
  Entraîner le réseau jusqu'à une solution raisonnable
  Calculer le critère de pertinence pour chaque poids
  Supprimer les poids les plus inutiles
jusqu'à *Critère sur le nombre de poids respecté*

ALGORITHME 2 – Principe de l'OBD [Le Cun et coll., 1990b] et de l'OBS [Hassibi et Stork, 1993]

relativement rapide. Pour reprendre le vocabulaire de la compression d'images, on peut dire que les deux opérations sont largement asymétriques. L'intérêt d'un système de reconnaissance réside dans l'obtention d'une solution performante dans des délais raisonnables quitte à nécessiter un processus d'apprentissage long.

En plus de ces considérations générales propres aux réseaux neuronaux, il ne faut pas perdre de vue que dans le cas du réseau perceptif, outre les problèmes évoqués plus haut, il faut aussi tenir compte de l'allongement du temps d'exécution dû aux cycles perceptifs eux-mêmes. Il y aura autant de propagation dans le réseau que de cycles perceptifs à effectuer pour reconnaître toutes les formes.

Dans notre construction du RNP, la topologie est suffisamment simple (des PMC sans couche cachée) et traite au final un nombre raisonnable de neurones et de poids (moins de 1 500) pour que l'apprentissage et la reconnaissance soient rapides. Ils le sont en effet si et seulement si l'on prend en considération uniquement le temps mis par la propagation et le retour de contexte à l'intérieur du réseau. Dans notre cas, le goulot d'étranglement provient essentiellement du calcul des entrées à fournir et non pas de l'utilisation du réseau lui-même. Le problème auquel nous sommes confrontés est de nature différente de ceux que l'on retrouvera dans la littérature.

Étant donné que tous les indices physiques formant le vecteur d'entrée sont issus de l'image du document, le temps mis pour chaque extraction est beaucoup plus important comparé à celui mis par le réseau seul. Pour donner un exemple chiffré, si tous les indices physiques sont extraits d'une page de document, le temps de l'analyse physique peut varier de 4 à 30 secondes sur un ordinateur de bureau récent.

Ce temps élevé s'explique en particulier par le passage de l'OCR qui est l'outil permettant l'analyse physique de la page (texte brut et styles du document). Si l'image est de bonne qualité (bien orientée, 300 dpi au minimum et aucun bruit), il n'est pas rare d'atteindre la borne inférieure[10]. Dans le cas contraire, en plus de l'augmentation du taux d'erreur sur la structure physique, le temps d'analyse dépasse assez souvent la barre de la demi-minute.

[10]Tests sur six pages d'un article scientifique «propre», couleur, à 300 dpi sur un P4 3.6GHz
Temps de reconnaissance (hors temps chargement image et sauvegarde des résultats)
Omnipage 15 : 22 sec, Omnipage 14 : 20 sec, Finereader 7 : 41 sec
Temps moyen : moins de 5 sec par page.

4.3 Méthodes diminuant la taille de l'entrée

Différentes méthodes sont possibles pour diminuer la taille de l'entrée. Il existe les méthodes de réduction qui transforment les variables d'origine x_i en de nouvelles X_i contenant chacune de l'information provenant des x_i, et d'autres qui ne sélectionnent qu'un sous-ensemble Y des variables d'origine x_i (Tab. 4.1). Nous allons développer plus en détail ces deux grandes familles en nous intéressant plus particulièrement à la sélection de sous-ensembles dont nous justifierons le choix en S.-Sec. 4.4.2.

Réduction de variables	Sélection de sous-ensembles de variables
$\{x_1, \ldots, x_p\}$ les variables d'origine	
$\{X_1, \ldots, X_q\}$ les nouvelles variables ($q \ll p$)	
$\forall i \in [\![1, q]\!], X_i = f(x_1, \ldots, x_p)$	$\forall i \in [\![1, q]\!], Y_i = x_{f(i)}, f(i) \in [\![1, p]\!]$

TABLEAU 4.1 – Différence entre réduction et sélection de variables

4.3.1 La sélection de variables

Les méthodes de sélection de variables sont généralement classées en deux catégories : les méthodes par filtre *(filter methods)* et les méthodes embarquées. Les méthodes par filtre, pour les plus simples d'entre-elles, consistent à assigner un score d'utilité à chaque variable pour ensuite sélectionner les meilleures. Les méthodes embarquées utilisent quant à elles le prédicteur pour trouver le bon ensemble de variables.

La sélection de variables est un thème de recherche très actif qui s'applique parfaitement aux domaines dont la taille des entrées est grande. Les objectifs des méthodes de sélection sont triples [Blum et Langley, 1997] :

- accroître les performances qualitatives des prédicteurs ;
- rendre plus rentable le prédicteur par une diminution de son temps d'exécution ;
- mieux comprendre les raisonnements sous-jacents qui ont permis de générer les sorties du prédicteur.

En comparaison à la réduction de variables (S.-Sec. 4.3.4), nous disposons ici d'une méthode qui répond exactement à ce que nous recherchons pour les cycles perceptifs, à savoir une diminution du temps d'exécution et une meilleure transparence du raisonnement du système. Sans celle-ci, les cycles perceptifs sont difficilement praticables pour ne pas dire impossibles.

4.3.2 Classement de variables

Le classement de variables *(variable ranking)* est la méthode la plus utilisée pour sa simplicité et ses bons résultats dans les expérimentations, bien que, d'un point de vue théorique, elle semble être a priori la moins bonne des solutions.

En effet, considérons un ensemble de m échantillons comme nous les manipulons dans les bases d'apprentissage pour les réseaux de neurones. Cet ensemble est composé de couples $\{x_k, y_k\}$ où x_k est un vecteur d'entrée et y_k la sortie attendue. Le classement de variables consiste simplement à utiliser une fonction de score S qui prend en compte chaque composante i du vecteur x d'entrée et la sortie souhaitée y pour déterminer l'importance d'une variable. Un score $S(i)$ élevé indiquera que la variable i est importante. En prenant les variables par ordre décroissant de $S(i)$, on obtient un ensemble contenant les plus informatives.

Les fonctions de score proviennent généralement d'outils statistiques comme le coefficient de corrélation C :

$$C(i) = \frac{\sum_k (x_{k,i} - \bar{x_i})(y_k - \bar{y})}{\sqrt{\sum_k (x_{k,i} - \bar{x_i})^2 \sum_k (y_k - \bar{y})^2}} \tag{4.1}$$

ou le critère de Fisher F :

$$F = \frac{(\mu_1 - \mu_2)^2}{s_1^2 + s_2^2} \tag{4.2}$$

avec μ la moyenne et s la dispersion. D'autres mesures sont envisageables [Torkkola, 2003]. La plupart d'entres-elles se basent sur une estimation empirique de l'information mutuelle entre chaque variable et la sortie *(information theoric criteria)* :

$$T(i) = \int_{x_i} \int_y p(x_i, y) \log \frac{p(x_i, y)}{p(x_i)p(y)} dxdy \tag{4.3}$$

avec $p(x_i)$, $p(y)$ les densités de probabilité de x_i et y et $p(x_i, y)$ la densité jointe.

L'inconvénient majeur de ce type de méthode vient du fait que l'ensemble de variables reste sous-optimal. En effet, choisir les q meilleures variables ne garantit pas de construire le meilleur ensemble de q variables. En effet, il est possible que dans l'ensemble de variables créé, il en existe certaines qui soient redondantes ou qui portent pratiquement la même information. Dans des cas extrêmes, on pourrait imaginer choisir au final q variables x_i avec, pour n'importe quel échantillon, $x_1 = x_2 = \ldots = x_q$. L'ensemble de variables serait alors composé de n fois la meilleure variable et ne serait porteur au bout du compte que d'une seule information.

Sans aller jusqu'à cette configuration théorique, il est quand même probable dans notre cas que les variables à traiter soient parfois dépendantes les unes des autres jusqu'à être linéairement liées (ex : la hauteur d'une ligne de texte et la taille de la police). La méthode du classement est donc dans la pratique un assez mauvais choix car nous utilisons un réseau de neurones pour la classification ; le fait de donner un ensemble de variables en entrée où la redondance est très forte ne permettra pas au réseau d'obtenir de bons résultats.

Un autre phénomène, pour ainsi dire «inverse», est aussi envisageable : une variable peut paraître inutile en elle-même mais s'avérer informative lorsqu'elle est prise avec d'autres. L'exemple du XOR illustre une telle configuration (Fig. 4.1) : on construit quatre exemples pour deux classes que l'on place aux quatre coins du carré unitaire et on les répartit selon la fonction logique XOR. Si l'on observe uniquement les projections sur les axes, il n'y a pas de séparation possible. Si l'on se place maintenant dans un espace à deux dimensions, les classes sont facilement séparables (avec une fonction de décision non linéaire).

Construire un ensemble de variables par la méthode du classement est donc en théorie une pratique assez risquée qui a comme conséquence des effets néfastes pour des classifieurs comme le Perceptron multicouche. Ce type de sélection a plus d'intérêt quand le nombre de variables est très grand et que la sélection doit s'effectuer rapidement.

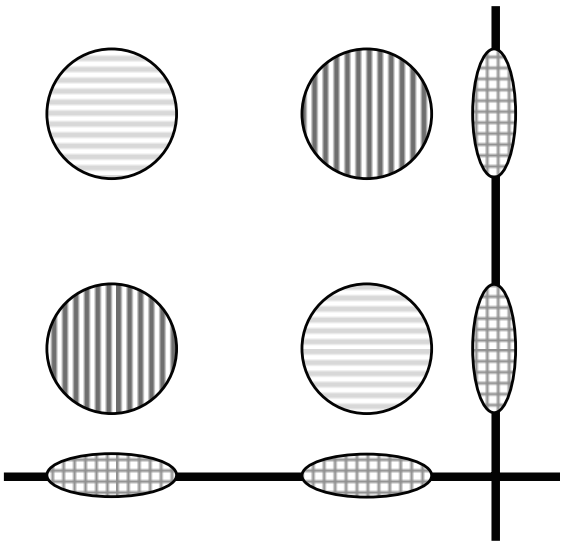


FIGURE 4.1 – Le cas du XOR : les deux axes n'apportent aucune information à eux seuls. Pris conjointement, ils permettent de séparer le problème

Dans notre cas, l'objectif de la sélection de variables sera de produire un groupe discriminant et composé de préférence de variables complémentaires plutôt que redondantes. Pour atteindre ce but, il est préférable de ne pas traiter indépendamment les variables mais de les considérer ensemble et surtout de ne pas évaluer la qualité propre de chacune mais la qualité d'un groupe formé par plusieurs variables. La méthode à retenir s'orienterait alors plus logiquement vers des techniques de sélection d'ensembles de variables que nous allons décrire dans la sous-section suivante.

4.3.3 Sélection de sous-ensembles de variables

Nous avons montré dans la section précédente à la fois les avantages calculatoires du choix d'une méthode de type classement de variables ainsi que les inconvénients théoriques et pratiques qu'il est probable de rencontrer lors de sa mise en application. Il paraît donc plus efficace d'avoir recours à une méthode se concentrant sur l'intérêt d'un groupe de variables plutôt que sur un calcul de pouvoir informatif considérant une seule variable à la fois.

Il existe trois familles de techniques dans la littérature dont la finalité est d'effectuer une sélection de sous-ensembles de variables *(variable subset selection)*. Il y a d'un côté les méthodes à adaptateur *(wrapper methods)* qui utilisent le système de classification comme une boîte noire uniquement pour évaluer le pouvoir prédictif d'un groupe de variables. Dans un esprit similaire, la famille des méthodes embarquées *(embedded methods)* se servent également du classifieur mais vont sélectionner les variables durant la phase d'apprentissage. De l'autre côté, les approches par filtre *(filter methods)* ont une philosophie inverse qui consiste à effectuer la sélection indépendamment du classifieur pendant une étape de prétraitement.

Méthodes à adaptateur

Les méthodes à adaptateur sont très utilisées ces dernières années. Le principe de base est le suivant : on considère déjà acquis le classifieur et, sans aucune information a priori à son sujet, il est utilisé tel quel pour qualifier l'utilité d'un groupe de variables (Fig. 4.2). La difficulté de la méthode provient de la génération de ces groupes qui doit être renouvelée pour obtenir des ensembles plus performants à chaque itération. Le choix de la fonction objectif et du critère d'arrêt sont eux aussi des problèmes difficiles à résoudre car ils sont la clé pour l'obtention d'une solution satisfaisante générée en un minimum d'itérations.

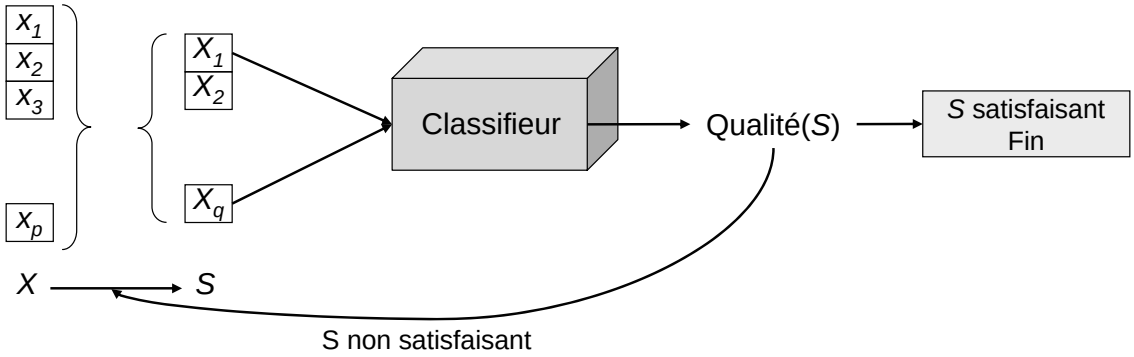


FIGURE 4.2 – Principe général d'une méthode à adaptateur

Pour disposer d'une méthode exacte (c'est-à-dire obtenir l'optimum), il serait toujours envisageable d'énumérer toutes les combinaisons de sous-ensembles et de les tester à travers le classifieur. Dans un cas pratique, sauf si l'on considère un nombre très restreint de variables, le problème est de classe *NP* et n'est donc pas réalisable. On utilise donc plus généralement des stratégies standard pour réduire le nombre de sous-ensembles à générer par des techniques de type séparation et évaluation *(branch and bound)*, de recuit simulé, ou bien encore des méthodes basées sur des algorithmes génétiques [Kohavi et John, 1997]. Ces derniers sont d'ailleurs les plus puissants et les plus utilisés depuis quelques années. Dans [Kim et Kim, 2000 ; de Oliveira et coll., 2001] les auteurs montrent comment les adapter dans le cas de la reconnaissance de chiffres manuscrits, la solution proposée est même renforcée dans [de Oliveira et coll., 2006] avec l'utilisation d'une version multiobjectif.

Les méthodes à adaptateur sont des méthodes assez simples à mettre en œuvre et donnant de bons résultats. Le côté «force brute» peut être toutefois un frein à leur utilisation ; si le fait de considérer le classifieur comme une boîte noire permet de conserver une sorte d'universalité du résultat obtenu, le nombre de combinaisons à tester est souvent très important et ne donne en général pas de meilleurs résultats qu'une méthode embarquée.

Méthodes embarquées

Les méthodes embarquées choisissent le groupe de variables durant l'apprentissage du classifieur. La recherche du meilleur sous-ensemble est guidée par l'apprentissage avec par exemple la mise à jour de la fonction objectif. Des techniques déjà évoquées [Le Cun et coll., 1990b] peuvent être utilisées pour supprimer les variables non pertinentes.

Le problème des méthodes embarquées est lié au fait qu'il faille justement adopter une stratégie en adéquation avec le type du classifieur : l'*Optimal Brain Damage* ne s'applique qu'aux Perceptrons multicouches. Pour les SVM (*Support Vector Machines*, séparateurs à vastes marges [Vapnik, 1995]), le procédé d'orthogonalisation de Gram-Schmidt sera préféré [Stoppiglia et coll., 2003].

D'autres approches allient à la fois le pouvoir prédictif du sous-ensemble et le nombre de variables utilisées. En maximisant le premier et en minimisant le second, l'ensemble créé est encore plus intéressant pour le classifieur car il apporte un très bon rapport entre qualité et rapidité d'extraction [Weston et coll., 2003].

Méthodes par filtre

Les méthodes par filtre, comme évoquées dans le cas du classement de variables, ont l'avantage d'être les plus rapides malgré les progrès en la matière des dernières méthodes embarquées. Un autre argument en faveur des méthodes par filtre vient de l'aspect générique de la solution proposée : la sélection de variables ne s'effectue ni pour ni par un classifieur à apprentissage. Le filtrage, comme son nom l'indique, s'utilise en phase de prétraitement ce qui permet de réduire à la fois la dimension des entrées et de se prémunir dans certains cas du phénomène de surapprentissage.

La simplicité et l'efficacité de l'approche par filtre est souvent mise en avant, comme par exemple dans [Bi et coll., 2003] où, bien qu'un SVM non linéaire soit utilisé en tant que classifieur, la méthode de sélection se fait elle aussi par un SVM mais linéaire et en amont de la reconnaissance.

D'autres systèmes neuronaux comme le PMC sont utilisés pour effectuer une sélection. Dans [Rivals et Personaz, 2003], les auteurs tentent d'éliminer les variables inutiles, le procédé se déroulant en deux phases : la première additive qui entraîne des PMC candidats avec un nombre croissant de neurones dans les couches cachées, puis, les candidats obtenus, une seconde phase de sélection, par des tests de Fisher, est effectuée sur les réseaux. Le processus stoppe dès l'obtention du plus petit réseau dont les variables et les neurones cachés ont une contribution significative pour le problème de régression. Une discussion sur le même problème est donnée par [Stoppiglia et coll., 2003].

Les méthodes par filtre ne reposent généralement pas sur des méthodes aussi complexes que celles proposées précédemment, ce sont souvent des méthodes statistiques comme celles présentées en sous-section 4.3.2 qui sont privilégiées. Dans [Hall et Smith, 1997], les auteurs proposent, pour accroître les performances de plusieurs classifieurs à apprentissage, de sélectionner via une heuristique de sélection par corrélation, un sous-ensemble de variables pour les classifieurs. L'heuristique tient compte de l'utilité individuelle de chaque variable et de sa corrélation avec les autres variables. Dans le même ordre d'idée, [Yu et Liu, 2003] proposent aussi une heuristique, nommée FCBF (pour *fast correlation-based filter*) qui introduit la notion de corrélation de prédominance qui a l'avantage d'avoir une complexité quasi linéaire au lieu d'une complexité au moins quadratique comme il est fréquent d'en rencontrer dans les méthodes se basant sur la corrélation.

4.3.4 Réduction de données

La réduction de données, ou reconstruction de données, a aussi pour but de réduire le nombre de variables à fournir en entrée d'un classifieur. Contrairement aux méthodes de sélection vues

précédemment, la réduction ne choisit pas un lot de variables mais les transforme en un autre ensemble de taille plus réduite. Mathématiquement, le problème se pose de la façon suivante : étant donné une donnée x de dimension p, comment lui trouver une représentation s de dimension k avec $k \leqslant p$ qui capture au mieux le contenu de la donnée x suivant un certain critère ? On appelle parfois les composantes de s les composantes cachées.

Les techniques couramment employées sont de type linéaire : les composantes de la nouvelle donnée sont des combinaisons linéaires des composantes d'origine :

$$\forall i \in [\![1, k]\!], s_i = w_{i,1}x_1 + \ldots + w_{i,p}x_p \tag{4.4}$$

ou, mis sous forme matricielle :

$$s = Wx, \quad \text{avec } W_{k\times p} \text{ la matrice de poids permettant la transformation linéaire} \tag{4.5}$$

Il y a de nombreux ouvrages et articles traitant de la réduction (linéaire et non linéaire) car elle couvre de vastes domaines dont les plus demandeurs sont les statistiques, le traitement du signal et aussi l'apprentissage automatique. Les méthodes les plus connues sont l'analyse en composantes principales, la poursuite de projection, les courbes principales, les cartes auto-organisatrices ainsi que certains réseaux de neurones et l'analyse en composantes indépendantes [Carreira-Perpiñán, 1997 ; Hyvärinen, 1999].

Nous montrerons à la sous-section 4.4.2 que les méthodes de réduction sont inappropriées à l'objectif que nous nous fixons. Il est cependant intéressant d'introduire ici les principes de l'analyse en composantes principales dont une partie sera utilisée dans notre méthode de partitionnement.

L'analyse en composantes principales (ACP) est la meilleure technique de réduction linéaire de dimension au sens des moindres carrés [Jolliffe, 2002]. C'est une méthode de second ordre basée sur la matrice de covariance des variables. Pour être plus précis dans le vocabulaire, l'ACP est l'ensemble des étapes de la méthode permettant le passage des données d'origine aux données dans l'espace réduit. La formule permettant de trouver la nouvelle base est un procédé d'algèbre linéaire de décomposition en valeurs singulières ou SVD *(Singular Value Decomposition)*. Suivant les domaines, elle est aussi connue comme la transformée de Karhunen-Loève ou d'Hotelling ou bien encore la méthode de la fonction orthogonale empirique EOF *(Empirical Orthogonal Function)*.

L'ACP réduit la dimension des données en trouvant des combinaisons linéaires orthogonales (les composantes principales) de ces données d'origine avec la plus grande variance. La première combinaison $s_1 = x^T w_1$ est la combinaison linéaire avec la plus grande variance, la deuxième combinaison a la seconde plus grande variance mais orthogonale à la première, et ainsi de suite. Pour de nombreuses bases de données, les premières composantes principales expliquent la plus grande partie de la variance tandis que les autres peuvent être négligées de part la faible information quelles portent (Fig. 4.3).

Si x_i est un vecteur de caractéristiques de $\mathbb{R}^p$ centré et réduit ($\mu_{x_i} = 0$ et $\sigma_{x_i} = 1$) et que la base est composée de n échantillons, on peut établir la matrice des observations X définie par :

$$X = \{x_{i,j}\}, i \in [\![1, p]\!], j \in [\![1, n]\!] \tag{4.6}$$

la matrice de covariance Σ_X s'écrit :

$$\Sigma_X = \frac{1}{n} X X^T \tag{4.7}$$

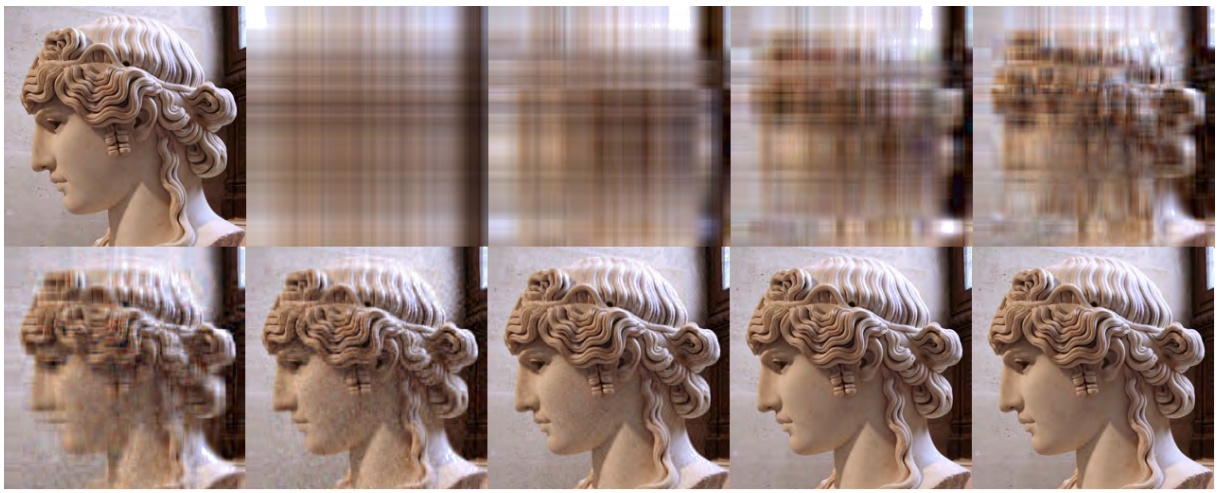


FIGURE 4.3 – Approximations successives d'une image RVB (original en haut à gauche) par la SVD, avec 1, 2, 4, 8, 16, 32, 64, 128 puis toutes les valeurs singulières (en bas à droite)

on peut décomposer Σ_X sous la forme :

$$\Sigma_X = U\Lambda U^T \tag{4.8}$$

avec $\Lambda = diag(\lambda_1, \ldots, \lambda_p)$ la matrice diagonale contenant les valeurs propres en ordre décroissant $\lambda_1 \geqslant \ldots \geqslant \lambda_p$ et U une matrice orthogonale $p \times p$ contenant les vecteurs propres. Les composantes principales sont données par les p lignes de la matrice S définie par :

$$S = U^T X \tag{4.9}$$

la matrice W du début de sous-section est donnée par U^T.

Pour le moment, les variables d'origine sont écrites dans une nouvelle base de même dimension que celle d'origine, la véritable réduction s'effectuant lorsque l'on choisit le sous-espace engendré par les k premiers vecteurs propres qui est le meilleur sous-espace de dimension k pour reconstruire X au sens des moindres carrés. La proportion de variance expliquée V_k par les k premières composantes principales est :

$$V_k = \sum_{i=1}^{k} \frac{\lambda_i}{trace(\Sigma_X)} \tag{4.10}$$

Trouver le nombre k n'est pas trivial et dépend des attentes de l'utilisateur, si son but est de «compresser» le plus possible les données, un k petit est suffisant et le taux de compression est simplement donné par $\frac{k}{p}$. S'il veut expliquer le maximum de variance, il doit choisir un k de telle sorte que $\sum_{i=1}^{k} \lambda_i$ dépasse un certain seuil qui lui semblera convenable.

D'une manière générale, l'interprétation des composantes principales est très difficile. Bien que les nouvelles variables construites par combinaison linéaire des variables d'origine soient décorrélées et aient des propriétés intéressantes, elles ne correspondent pas forcément à des quantités physiques interprétables [Balci et Atalay, 2002 ; Etemad et Chellappa, 1997]. Il est possible d'émettre des suppositions et de dégager quelques principes génériques (Tab. 4.2), mais en aucun cas des règles établies.

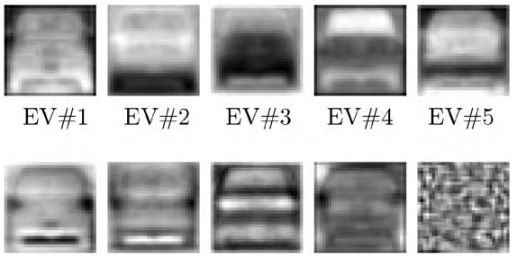
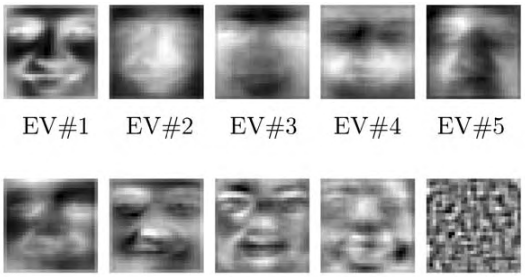
Échantillons de la base	Quelques vecteurs propres
	
	

TABLEAU 4.2 – Dans une image, les premiers vecteurs propres semblent encoder l'information lumineuse, les suivants semblent plutôt encoder les caractéristiques locales, les derniers sont souvent considérés comme représentatifs de bruit [Sun et coll., 2004]

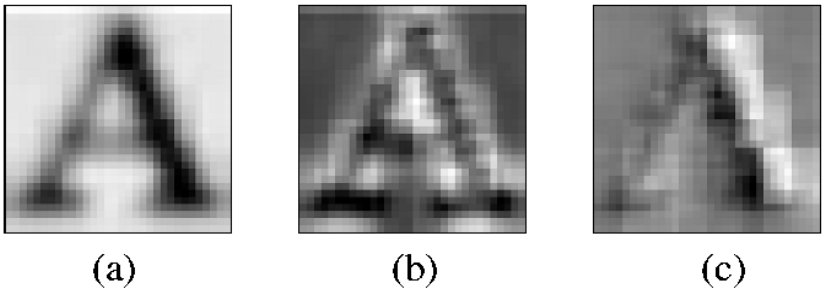


FIGURE 4.4 – Images des vecteurs propres pour le caractère 'A' [Yanadume et coll., 2004]. L'image (a) représente le premier vecteur propre, (b) et (c) respectivement le deuxième et troisième vecteur propre

4.4 Partitionnement de l'espace d'entrée

4.4.1 Contraintes sur le choix de la méthode à proposer

Dans notre cas, il n'est pas possible d'utiliser une méthode de réduction d'espace. En effet, bien que ce type de méthode donne généralement de meilleurs résultats qu'une simple sélection des variables d'entrée, les méthodes de réduction ont l'inconvénient de supprimer la correspondance entre les variables de l'espace de départ et celles de l'espace réduit.

La réduction d'espace peut être vue comme une compression destructive : une fois les données de départ compressées, il n'est plus possible de retrouver par décompression exactement les données d'origine. Dans le meilleur des cas, il est envisageable d'en avoir une approximation dont la qualité dépendra de la taille de l'espace réduit. En général, ce problème n'en est pas un réellement car la faible perte en terme de qualité est largement compensée par une très forte réduction de la complexité.

Pour le RNP, la perte de qualité n'est pas l'inconvénient majeur. En effet, le vrai problème vient du fait que tous les calculs se feront uniquement dans l'espace réduit et il sera alors impossible de répercuter des interprétations sur les variables de l'espace réduit vers l'espace d'origine.

Concrètement, si nous utilisons un schéma tel que le montre la figure 4.5, la réduction des données en entrée du réseau pourrait être bénéfique en temps de calcul et peut-être même en précision. Reste que la finalité du RNP est de corriger les données en entrée pour espérer affiner ses résultats lors d'une prochaine propagation avec les entrées corrigées.

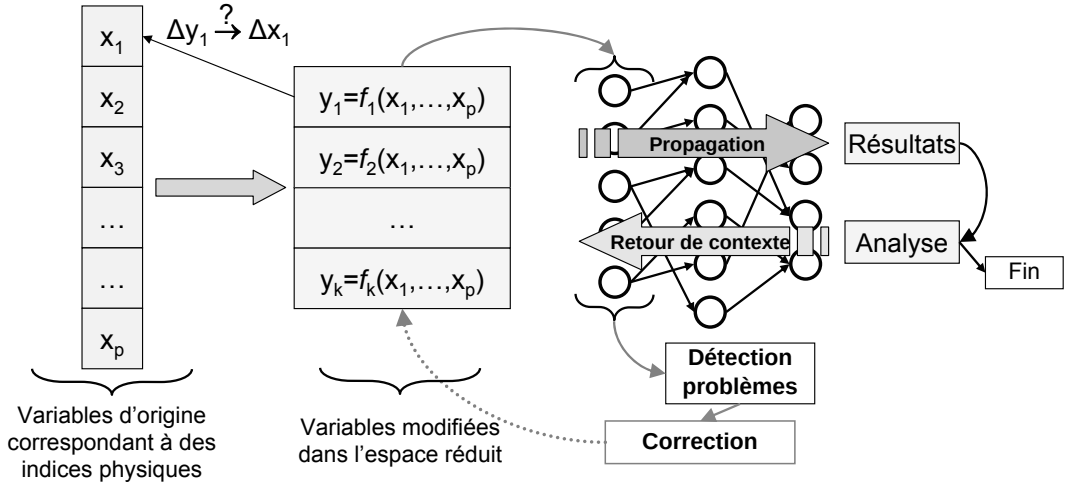


FIGURE 4.5 – Types de connexions entre les niveaux et les neurones

L'intérêt du RNP ne réside pas exclusivement dans l'obtention d'un bon résultat dès la première propagation : sa force vient des cycles perceptifs. Contrairement à ce qui est fréquemment vu dans la littérature, la technique que nous voulons mettre en œuvre ne consiste pas à obtenir uniquement les meilleurs paramètres d'un réseau. Nous voulons trouver un moyen de modifier et d'adapter les nouvelles observations après chaque cycle perceptif, ceci dans le but de mettre en relation les observations avec les conclusions déjà trouvées lors de l'apprentissage. Si nous décidons de garder une technique de réduction, les retours de contexte, tels que nous les avons décrits dans le chapitre 3, seront alors impossibles. En effet, même si l'on pouvait toujours émettre des hypothèses sur les entrées réduites en cas de mauvaise reconnaissance, celles-ci ne concerneraient que les données de l'espace réduit $(y_1, \ldots, y_k)$. De plus, même dans le cas classique d'une ACP avec des combinaisons linéaires f_i, il n'est pas possible de revenir aux données de départ x_i.

Sachant la correction à apporter à l'un des y_i, il n'est pas réalisable de répercuter précisément la variation Δy_i sur les x_i car la contribution de chaque x_i est perdue dans le calcul des y_i.

Même en corrigeant « mathématiquement » tous les x_i afin de redonner un nouveau vecteur y_i correspondant aux variations Δy_i, les variations Δx_i ne donneront pas de modification cohérente sur les variables provenant des observations physiques.

La seule solution possible est d'utiliser une méthode de sélection de variables. Plutôt que de réduire l'espace d'entrée en perdant les variables d'origine :

$$\begin{array}{c} E^n \rightarrow F^k \\ reduction: \quad X = (x_1, \ldots, x_p) \mapsto Y = (y_1, \ldots, y_k) \\ n \gg k, \forall i, y_i = f(x_1, \ldots, x_p) \end{array} \tag{4.11}$$

il est préférable de choisir un sous-ensemble des variables d'origine en entrée du système :

$$\begin{array}{c} E^n \rightarrow E^k \\ selection: \quad X = (x_1, \ldots, x_p) \mapsto Y = (y_1 = x_{u_1}, \ldots, y_k = x_{u_k}) \\ p \gg k, \forall i \exists j, x_{u_i} = x_j, \forall i \forall j, i \neq j \Rightarrow u_i \neq u_j \end{array} \tag{4.12}$$

Dans une méthode de sélection de données, on part donc toujours des variables d'origine à savoir dans notre cas des informations sur la structure physique du document.

4.4.2 Justification de la méthode

Si l'on reconsidère les méthodes de réduction, nous avons déjà critiqué le fait que l'information d'origine était diluée dans les nouvelles variables réduites et rendait de ce fait impossible les retours de contexte. Une autre limitation de taille s'avérant pénalisante pour nos cycles perceptifs vient du fait que, potentiellement, un y_i de l'espace réduit nécessite tous les x_i de l'espace d'origine car $y_i = f(x_1, \ldots, x_p)$. Nous avons montré, dans notre cas, que la réduction du nombre de variables n'influe que très peu sur la rapidité du fonctionnement interne du RNP ; c'est le temps passé à extraire les x_i qui est largement plus élevé. À titre de comparaison, si l'on se place dans le meilleur des cas, l'extraction de certains des x_i requiert au minimum un passage d'OCR sur la page à traiter et demanderait donc environ cinq secondes. D'un autre côté, une fois les x_i obtenus, le passage dans le réseau est immédiat : il faut moins d'une seconde pour propager 10 000 entrées. Le rapport le moins optimiste entre extraction et propagation est de 40 000 (l'extraction est 40 000 fois plus lente que la propagation). Dans un cas moyen réel, on peut considérer qu'un rapport de 100 000 correspond mieux aux expérimentations.

La sélection de variables permet donc de réduire ce temps d'extraction. Si le temps était identique pour chacune des variables, diviser la taille de l'entrée par trois nous ramènerait dans une configuration favorable à un cycle perceptif de l'ordre de la seconde. On peut aussi concevoir cette sélection comme la possibilité d'effectuer trois cycles perceptifs d'une durée semblable à un cycle sans sélection.

Nous avons parlé jusqu'à présent des atouts de la sélection uniquement au niveau du temps d'analyse tout en montrant l'inadéquation d'une méthode de réduction avec l'utilisation des cycles perceptifs. Nous n'avons pas encore évoqué les conséquences au niveau de la qualité des résultats. En effet, le fait de ne pas utiliser l'ensemble des variables conduit nécessairement à se priver d'information utile à la poursuite d'une analyse fine. Hormis les cas isolés où certaines variables n'apportent que du bruit ou des données erronées à cause d'une mauvaise extraction des observations, chaque variable a son importance. Même si l'importance isolée de la variable semble nulle, sa contribution avec d'autres peut donner une information plus pertinente (S.-Sec. 4.3.2). Ceci est d'autant plus vrai pour les réseaux de neurones qui donnent de meilleures classifications quand l'entrée est grande et variée.

La méthode que nous proposons n'est pas à proprement parler une sélection de variables mais plutôt un partitionnement de l'espace d'entrée :

$$E^p \rightarrow \bigcup_{i=1}^{q} E^{u_i}, \forall i, j, E^{u_i} \cap E^{u_j} = \emptyset \tag{4.13}$$

le principe étant de créer des ensembles disjoints à partir des variables d'origine. Chaque groupe sera donné progressivement au RNP en fonction de la difficulté de la forme à reconnaître : on débutera par un groupe de petite taille pour les premiers cycles perceptifs. Si la correction n'apporte rien à la reconnaissance d'une forme, on utilisera alors un autre groupe de données en complément du précédent pour apporter plus d'informations (Fig. 4.6).

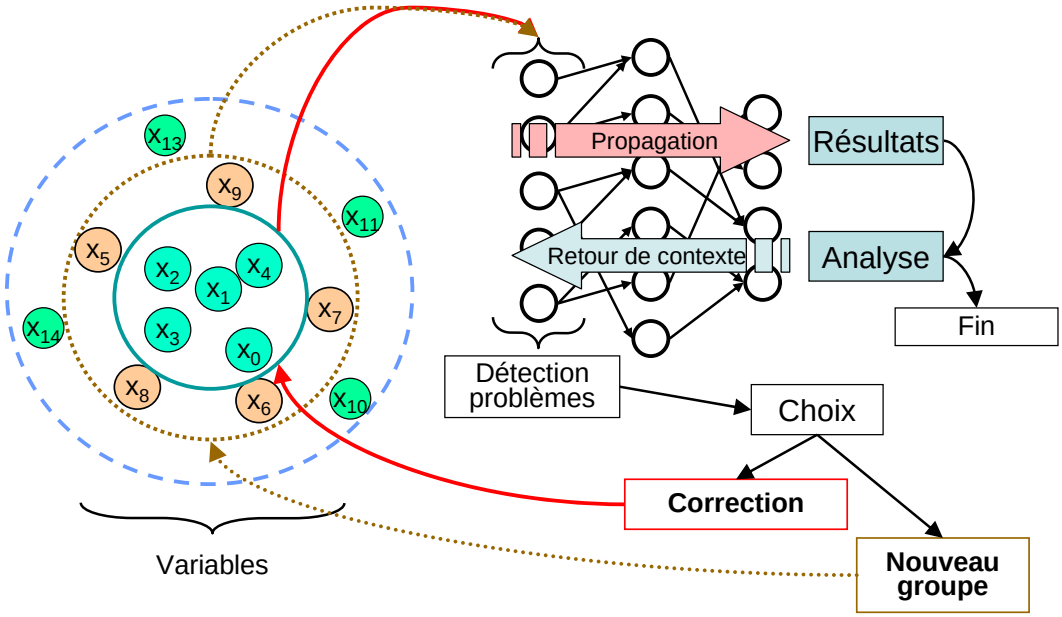


FIGURE 4.6 – Reconnaissance du RNP avec partitionnement de l'espace d'entrée

Le réseau de neurones perceptif muni de ce partitionnement mérite encore plus son qualificatif «perceptif» : en plus d'utiliser une approche mixte entre analyse descendante et ascendante, de corriger ses entrées par utilisation de contexte, le réseau utilise une approche progressive entre vision globale et vision locale. La vision globale peut s'apparenter au premier groupe de variables : il est restreint, peu informatif et très peu spécialisé sur chacune des formes à reconnaître, il donne une information approchée et générale mais qui peut suffire à reconnaître une partie des formes et faciliter la suite de l'analyse. La vision locale peut être rattachée à l'utilisation de l'ensemble des sous-ensembles de variables : si les premiers extracteurs ne sont pas suffisants pour labelliser une forme, des indices physiques plus fins et plus adaptés seront employés en fonction de la difficulté de l'analyse.

Pour construire les groupes de variables, deux méthodes sont envisageables. La première consiste à classer les variables par rapidité d'extraction, le premier groupe est composé des indices les plus simples à extraire, les autres groupes étant progressivement plus lents à obtenir. Construire de tels groupes est assez évident : on peut donner un score à chaque variable en

fonction de la complexité théorique de l'algorithme d'extraction permettant son obtention ou bien encore, plus simplement, en chronométrant le temps moyen mis par chaque extracteur. Si la construction est simple, les groupes de variables créés ne seront pas nécessairement de «bons» groupes pour un classifieur comme le RNP : la redondance et le faible pouvoir informatif global d'un groupe peuvent être malheureusement présents en même temps. Si l'on gagne en rapidité pour chacun des cycles perceptifs, il sera sûrement nécessaire d'effectuer de nombreux cycles additionnels pour atteindre un bon taux de reconnaissance, le temps global de reconnaissance pouvant être au final aussi élevé qu'en n'utilisant pas de partitionnement.

Une autre façon de procéder consiste à classer les variables par pouvoir informatif : les groupes pourraient être présentés au RNP par utilité décroissante. L'idée de donner un premier groupe de taille réduite et comportant une information presque identique à l'ensemble d'origine est tout aussi judicieuse pour réduire le temps global de reconnaissance ; l'extraction nécessiterait plus de temps, mais il faudrait en contrepartie moins de cycles perceptifs.

Si la création d'un partitionnement par rapport à la rapidité d'extraction peut se faire très simplement, le partitionnement se basant sur le pouvoir discriminant des variables est loin d'être trivial comme nous l'avons montré dans les sections précédentes. Il faut aussi tenir compte du fait que le classifieur est un réseau de neurones ; les ensembles de variables doivent, si possible, contenir le moins de redondance, comme vu en sous-section 4.3.2, le groupe des n meilleures variables n'est pas forcément le meilleur groupe de n variables pour le classifieur.

À la lumière des différentes familles et méthodes de sélection de variables, nous avons opté pour une méthode à base de filtre, qui nous évitera de construire des RNP pour faire la sélection, la méthode sera aussi plus générique et pourra être considérée comme un prétraitement léger lors de l'élaboration du système de reconnaissance. Dans la prochaine sous-section, nous montrerons comment utiliser une méthode par filtre dont la première phase est empruntée à une méthode de réduction de données et qui nous donnera des groupes de variables disjoints ayant une faible redondance à l'intérieur de chacun.

4.4.3 Algorithme de la méthode

Nous avons basé une partie de la méthode de partitionnement sur l'analyse en composantes principales (S.-Sec. 4.3.4) pour ses propriétés mathématiques très intéressantes. En reprenant les mêmes notations, on dispose d'une matrice d'observations X dont chaque colonne est un échantillon centré de notre base d'apprentissage. Elle a p lignes, p étant la dimension de chaque échantillon. La matrice de covariance est $\Sigma_X = \frac{1}{n} X X^T$ avec n le nombre de colonnes de X.

En utilisant la méthode QR (voir Annexe E, page 153), on décompose Σ en valeurs propres et vecteurs propres $\Sigma_X = U \Lambda U^T$, Λ la matrice diagonale contenant les valeurs propres en ordre décroissant et U dont les colonnes sont les vecteurs propres correspondant aux valeurs propres.

Si l'on continuait la méthode de l'ACP, on pourrait écrire la nouvelle matrice Y d'observations en utilisant U_k la sous-matrice de U contenant uniquement les k premiers vecteurs propres :

$$Y = U_k^T X \tag{4.14}$$

et en utilisant la propriété $\forall k, U_k^T U = I_k$, si l'on veut approcher par $\tilde{X}$ la matrice d'observations d'origine, on a :

$$\tilde{X} = U_k Y \tag{4.15}$$

Si l'on veut maintenant sélectionner des variables de X tout en gardant les propriétés de l'ACP, il faudrait déterminer une permutation des lignes de U'_k telle que :

$$U'_k = \begin{bmatrix} I_k \\ [0]_{(p-k).k} \end{bmatrix} \tag{4.16}$$

et trouver l'optimal revient à essayer toutes les permutations possibles, ce qui n'est pas réalisable lorsque k est grand.

Une autre solution consiste à utiliser la constatation [Jolliffe, 2002] qu'une forte valeur de l'$i^{\text{ème}}$ coefficient de l'une des composantes principales implique que l'élément x_i de X est dominant dans la composante correspondante. Si l'on choisit les variables correspondant au plus grand coefficient de chacune des k premières composantes principales, on peut obtenir un groupe de variables très informatif au sens de l'ACP. Le problème de la redondance reste toujours présent car les variables sont traitées indépendamment et des variables avec de l'information similaire peuvent être choisies.

Si l'on reprend la matrice U_q et que l'on note P_i les p lignes de cette matrice, chaque P_i représente la projection de la $i^{\text{ème}}$ variable dans l'espace réduit et chaque composante de P_i donne le «poids» de la $i^{\text{ème}}$ variable sur chaque axe de l'espace. Si deux variables sont fortement corrélées, elles doivent avoir des P_i semblables en valeur absolue. Le phénomène est encore plus accentué lorsque l'on choisit la matrice de corrélation C_X à la place de la matrice de covariance Σ_X. Le raisonnement inverse est aussi juste : deux variables indépendantes auront des vecteurs P_i éloignés. L'exemple suivant illustre la constatation sur les observations X :

$$X = \begin{pmatrix} 4 & 2 & -8 & 0 & 9 & 5 & -8 \\ 4 & 5 & 6 & 2 & -4 & 0 & 1 \\ -4 & 0 & 0 & 6 & 8 & 2 & 1 \\ -2 & -1 & 4 & 0 & -4 & -2 & 4 \end{pmatrix}$$

$$C_X = \begin{pmatrix} 1 & -0.56 & 0.32 & -0.99 \\ -0.56 & 1 & -0.74 & -0.52 \\ 0.32 & -0.74 & 1 & -0.30 \\ -0.99 & 0.52 & -0.30 & 1 \end{pmatrix}$$

$$\Lambda = \begin{pmatrix} 2.74 & 0 & 0 & 0 \\ 0 & 1.04 & 0 & 0 \\ 0 & 0 & 0.22 & 0 \\ 0 & 0 & 0 & 0.001 \end{pmatrix}$$

$$U = \begin{pmatrix} -0.54 & -0.43 & -0.10 & -0.72 \\ 0.51 & -0.40 & -0.76 & -0.04 \\ -0.41 & 0.67 & -0.63 & -0.00 \\ 0.53 & 0.46 & 0.15 & -0.70 \end{pmatrix} |U_{k=2}| = \begin{pmatrix} 0.54 & 0.43 \\ 0.51 & 0.40 \\ 0.41 & 0.67 \\ 0.53 & 0.46 \end{pmatrix}$$

$P_1 = \begin{pmatrix} 0.54 & 0.43 \end{pmatrix}$ et $P_4 = \begin{pmatrix} 0.53 & 0.46 \end{pmatrix}$ sont proches, cela signifie que x_1 et x_4 sont probablement corrélées. C'est le cas car on a $x_4 \approx -\frac{1}{2}x_1$.

Si l'on se base sur les P_i, on peut regrouper les variables qui sont corrélées ensemble et de même pour celles qui sont indépendantes. Pour créer un groupe à la fois informatif et avec le moins de redondance possible, l'idée consiste à regrouper les P_i en plusieurs ensembles disjoints en fonction de leur distance, chaque groupe contenant des vecteurs proches au sens de la norme euclidienne. En sélectionnant un vecteur dans chaque groupe, les variables correspondantes forment un nouvel ensemble de variables très peu corrélées les unes par rapport aux autres. Pour construire un groupe informatif, il suffit de choisir x_i correspondant au P_i le plus proche de chaque centre de classe car c'est celui qui représente au mieux la classe correspondante et qui est le plus éloigné de toutes les autres classes.

Dans l'exemple précédent, une clustérisation en deux classes regrouperait P_1, P_3 et P_4 dans un même cluster et laisserait P_2 seul. Un bon groupe de deux variables serait donc x_2 et x_4 car P_2 étant seul, le meilleur représentant est lui-même, le meilleur représentant du premier cluster serait P_4 si le centre du cluster est $\begin{pmatrix} 0.53 & 0.43 \end{pmatrix}$. Le second sous-ensemble de variables serait de facto x_1 et x_3. (Fig. 4.7)

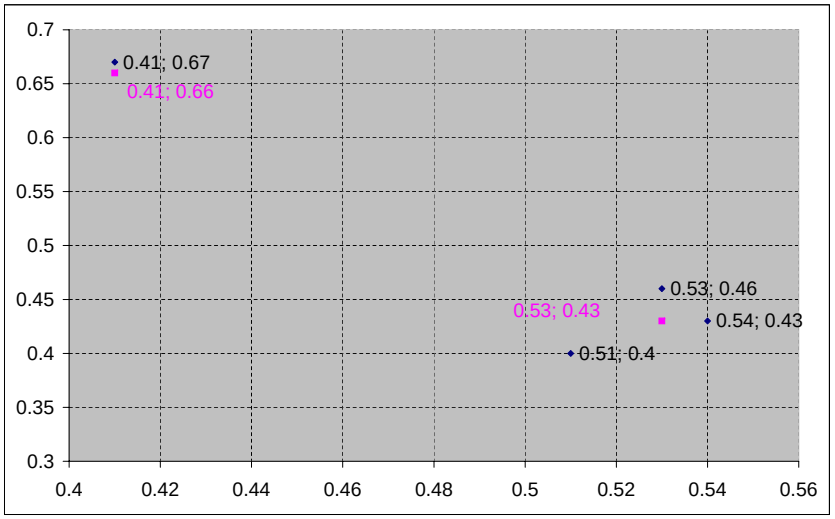


FIGURE 4.7 – Partitionnement des P_i

On notera aussi que d'autres comparaisons sont envisageables ; la distance euclidienne donne de bons résultats mais il serait très utile, dans un objectif de produire un meilleur premier sous-ensemble d'avoir recours à d'autres distances ou des mesures de similarité comme le cosinus qui semble être un bon candidat pour permettre une meilleure classification.

La méthode de sélection est donc simple à mettre en œuvre, la majorité des calculs se font à partir de la matrice de corrélation qui est de dimension $p \times p$, p étant la dimension d'une observation ; tous les algorithmes nécessaires à l'obtention du partitionnement sont cubiques en p. Un résumé des étapes à suivre est donné par l'algorithme 3 et une vue d'ensemble sur la méthode de sélection est présentée à la figure 4.8.

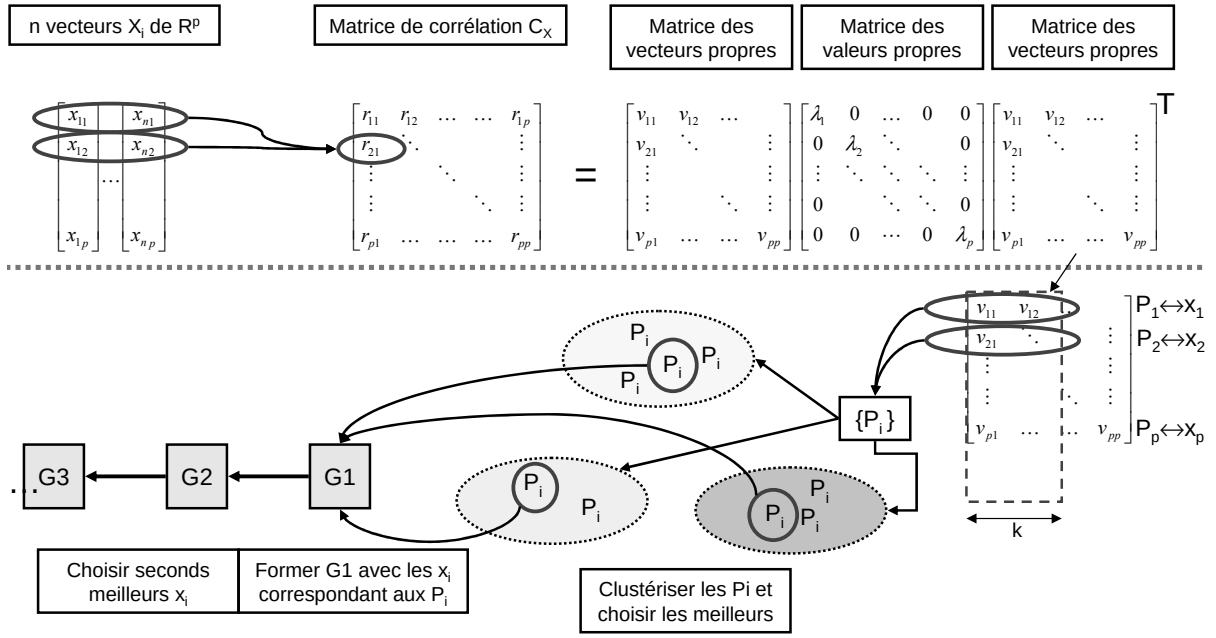


FIGURE 4.8 – Vue schématique de la méthode de partitionnement

Construire la matrice des n observations X de dimension p
Calculer la matrice de corrélation C correspondante à X
Effectuer une décomposition SVD de C avec par exemple la méthode QR telle que $C = U\Lambda U^T$, Λ la matrice diagonale des valeurs propres tirées et U la matrice des vecteurs propres correspondants
Choisir $k < p$ et U_k la sous-matrice de U contenant les k premiers vecteurs
Clusteriser les lignes P_i de $|U_k|$ en q clusters avec par exemple une SOM
$i \leftarrow 1$
répéter
  Choisir les q P_i les plus proches du centre de chaque cluster
  Former le $i^{\text{ème}}$ groupe G_i de variables x_j correspondant aux q vecteurs lignes P_i
  Supprimer ces q lignes P_i des clusters
  $i \leftarrow i + 1$
jusqu'à *reste des P_i dans les clusters*
Les G_i forment une partition triée des variables x_i d'origine par pouvoir prédictif décroissant et avec le moins de redondance possible à l'intérieur d'un G_i

ALGORITHME 3 – Résumé de la méthode de partitionnement

Il convient également de choisir le nombre final de sous-ensembles qui seront utilisés par le RNP. Étant donné le faible nombre de variables dont nous disposons, il nous a semblé que trois groupes étaient suffisants. D'une manière générale, si les groupes sont trop nombreux, il y

aura peu de nouvelles informations pour améliorer la reconnaissance de la forme ambiguë après chaque ajout de groupe. Dans l'absolu, nous avons montré que, sous réserve de disposer déjà de la valeur des entrées, la propagation et la correction dans le réseau étaient négligeables. On pourrait même insérer les variables les unes à la suite des autres (pour former un groupe de taille $n+1$ à chaque cycle). Du fait que nous considérions l'extraction d'indices physiques comme faisant aussi partie du temps de reconnaissance, il se peut que, suite à un changement de sous-ensemble, l'algorithme décide de corriger les variables. S'il fallait autant de cycles que de variables, avec le risque d'effectuer plusieurs fois des extractions, nous allongerions excessivement le temps de reconnaissance pour, au final, opérer substantiellement le même travail qu'en sélectionnant directement l'ensemble complet des variables lors du premier cycle. Pour qu'il y ait à la fois un gain de temps et des résultats proches d'une reconnaissance sans partitionnement, il faut des groupes équilibrés au sens du pouvoir informatif.

La méthode que nous préconisons est de choisir le nombre de clusters q du partitionnement comme étant la taille du premier groupe désiré avec si possible $q < k$, k étant la variance gardée dans la décomposition SVD. Le premier groupe est formé des q meilleures variables (G_1). Pour le deuxième groupe, on sélectionne parmi les groupes restants les deux meilleures variables de chaque cluster ($G_2 \cup G3$) et ainsi de suite pour que les clusters soient vidés de plus en plus vite. Sur nos 56 variables possibles, si le premier groupe en comporte 8, le deuxième en aura 24 (les 8 du premier et les 2×8 du courant), le troisième en aura 48 ($8 + 8 \times 2 + 8 \times 3$) que nous laisserons à 56 pour ne pas créer un quatrième groupe inutilement. Notons aussi que lorsque des variables de haut niveau sont utilisées, celles qui sont plus informatives sont aussi celles qui sont généralement les plus difficiles à extraire d'où l'intérêt de rajouter des ensembles de taille croissante pour permettre une extraction assez rapide dans les premiers cycles.

4.4.4 Choix de la dimension du sous-espace

Dans la méthode proposée, certains paramètres doivent être fixés par l'utilisateur pour poursuivre toutes les étapes de l'algorithme. Le paramètre k, indiquant la dimension de l'espace réduit, est le plus important dans les premières phases. Sa détermination est un problème récurrent pour toute solution utilisant une décomposition SVD et on retrouve nécessairement un grand nombre de discussions à ce sujet dans tout système employant une analyse en composantes principales [Hu et Xu, 2004].

Si toutes les valeurs propres sont retenues ($k = p$), la matrice de passage U_k est égale à U, ce qui équivaut à garder intacte toute l'information initiale. Les liaisons entre les variables ne sont pas davantage simplifiées et, dans le cas d'une ACP, on ne réduit pas l'espace des observations car X et Y ont les mêmes dimensions. Inversement, le fait de ne garder qu'un très faible nombre de valeurs propres ($0 < k \ll p$) n'explique qu'une partie minime de la variance totale. La compression sera certes grande, mais la complexité des liaisons entre les variables sera résumée de façon excessive et ne reflétera plus les tendances des variables d'origine (Fig. 4.3, p. 82). De manière générale, si les observations d'un phénomène complexe nécessitent une transformation telle que l'ACP pour être analysées plus facilement, il est alors rare que quelques valeurs suffisent à expliquer une proportion importante de la variance totale.

Trouver k optimal est un compromis entre la conservation et la simplification de l'information et se fait souvent de manière empirique (surtout dans des domaines comme les sciences humaines et sociales). Les trois principales façons de choisir k sont :

– fixer arbitrairement le nombre k. Nécessite une très bonne connaissance du phénomène étudié ;

- fixer un pourcentage de valeur à garder ($k = \frac{x \cdot p}{100}$). Similaire au cas précédent, on choisit les $x\%$ premières valeurs propres, seule une bonne connaissance du phénomène permet de trouver x ;
- choisir k de telle sorte que la somme des k premières valeurs propres dépasse un certain seuil. Cette méthode du pourcentage cumulé permet de définir une valeur connue de la variance totale, restera à déterminer quelle variance conserver pour fixer k.

Lorsque les données sont plus difficiles à interpréter, il existe des critères moins subjectifs, plus automatiques et robustes, basés sur la forme de la suite constituée par les valeurs propres :

- méthode de Kaiser qui calcule d'abord la moyenne de toutes les variances λ_i et qui choisit le premier k de manière à ce que la somme des k premières variances dépasse la moyenne ($min_k \sum_{i=1}^{k} \lambda_i > \frac{1}{p} \sum_{i=1}^{p} \lambda_i$) ;
- méthode de Cattell [Cattell, 1996] qui fixe k comme étant le « coude » de la fonction définie par les valeurs propres c'est-à-dire l'endroit où l'on observe une décélération de la décroissance de la fonction (Fig. 4.9). La méthode est aussi connue sous le nom de l'éboulis *(scree-test)* qui fait une analogie du choix de k avec un rocher s'effondrant du haut de la montagne formée par la courbe et stoppant sa course au pied de la montagne.

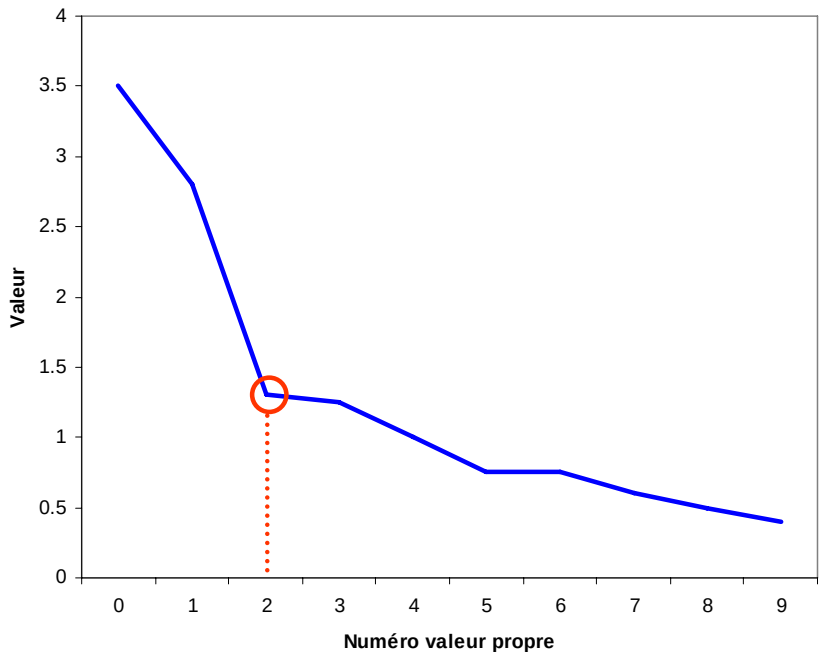


FIGURE 4.9 – Méthode de l'éboulis de [Cattell, 1996] : dans cet exemple, le coude se trouve à la troisième valeur propre

Ces deux derniers critères sont largement utilisés dans la littérature et permettent de justifier moins arbitrairement le choix de k. La méthode de Cattell est, dans notre cas, celle donnant en moyenne les meilleurs résultats.

4.5 Expérimentations

Nous avons effectué plusieurs tests pour fixer les paramètres de l'algorithme, la méthode de partitionnement a été employée comme un outil de sélection de variables [Rangoni et Belaïd, 2006]. Pour évaluer l'efficacité de la méthode et l'influence du paramètre k, deux bases ont été employées : la première est une base de chiffres manuscrits, la base MNIST (Annexe B) et la deuxième est celle des attributs physiques extraits des images de documents (Annexe A).

Tests sur des images de caractères manuscrits

La base est composée d'images en niveaux de gris, carrées, de 28 pixels de côté, les vecteurs de caractéristiques ont donc 784 composantes. Pour évaluer la qualité d'un groupe, on utilise un PMC comme classifieur ; son résultat de bonne reconnaissance sur une base de 10 000 images sera le score de qualité. La méthode de partitionnement est employée pour former des groupes de différentes tailles et elle est opposée à une autre sélection plus triviale consistant à retenir le groupe donnant les meilleurs résultats parmi un tirage aléatoire de 1 000 groupes. Le PMC a bien sûr les mêmes paramètres (topologie, initialisation des poids, etc.) pour chaque expérience.

Le tableau 4.3 montre les résultats obtenus suivant différentes valeurs de taille de groupes de variables (la méthode de Cattell étant utilisée pour le choix de k).

Nb. de variables	Méthode	
	Aléatoire	Partitionnement
784 (max)	100%	100%
500	98,4%	99,2%
300	95,9%	98,4%
150	90,5%	96,5%
100	84,2%	94,2%
50	70,9%	87,8%
25	47,1%	67,6%

TABLEAU 4.3 – Résultats de reconnaissance sur la base MNIST en fonction de différentes tailles de sous-ensembles de variables. Les résultats sont normalisés en rapport avec le taux atteint pour l'ensemble complet des variables

Les résultats sont normalisés suivant le meilleur taux de reconnaissance pouvant être obtenu avec l'ensemble complet des variables. On observe que le premier groupe formé par la méthode de partitionnement est toujours meilleur que la méthode aléatoire. En ne choisissant que 50 variables parmi les 784 possibles, 87.8% de l'information est conservée pour le PMC et malgré l'influence de chaque pixel sur le classifieur, on arrive à capter plus des deux tiers de l'information en gardant moins de 4% des variables. La figure 4.10 donne la répartition spatiale des pixels retenus par la méthode, la première image est l'image moyenne de la base ($\frac{1}{n}\sum_{i=1}^{n} I_i$), les autres étant les pixels faisant partie de l'ensemble créé par l'algorithme. On remarquera que les pixels ne sont pas forcément tous présents là où ils pourraient être attendus car le classifieur s'aide autant des points blancs que des points noirs pour décider d'une forme. Les pixels du bord ne

sont généralement pas sélectionnés, car les chiffres sont tous centrés dans l'image et ne touchent quasiment jamais les bords.

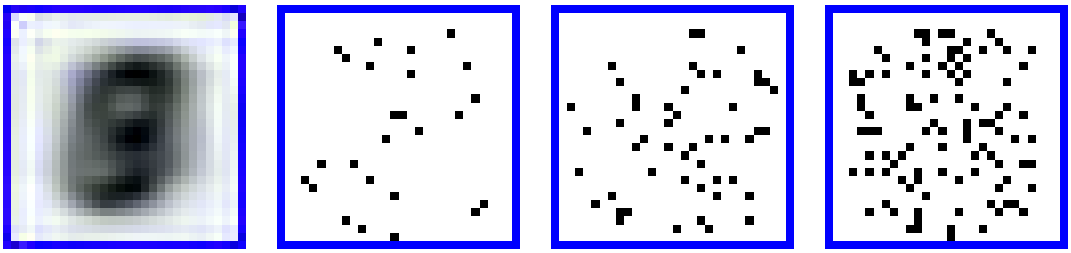


Image moyenne | 25 variables | 50 variables | 100 variables

FIGURE 4.10 – Localisation des variables de sous-ensembles de différentes tailles par la méthode de partitionnement

Le tableau 4.4 donne les résultats de la même expérimentation mais cette fois-ci sur des images réduites dans une matrice de 7×7 pixels. Les constatations sont les mêmes, le partitionnement donne toujours de meilleurs résultats, la conservation de l'information est moins importante (70% de l'information avec 30% des pixels), ce qui est tout à fait normal vu la taille de l'image dans laquelle chaque pixel a désormais son importance pour la classification.

Nb. de variables	Méthode	
	Aléatoire	Partitionnement
49 (max)	100%	100%
35	94,2%	99,3%
25	81,2%	88,6%
15	56,2%	70,5%
10	43,9%	55,2%

TABLEAU 4.4 – Résultats normalisés de reconnaissance sur la base MNIST redimensionnée en fonction de différentes tailles de sous-ensembles de variables

Tests sur des caractéristiques extraites de structures physiques

Après avoir montré des résultats sur une base de caractéristiques de bas niveau, nous présentons maintenant des expérimentations similaires mais sur des variables de haut niveau, leur valeur étant extraite par des outils travaillant quant à eux sur une image de document. La base, décrite plus en détail dans l'annexe A, comporte 74 documents découpés chacun en plusieurs blocs desquels sont extraites 56 caractéristiques physiques (Tab. 3.1, p. 55) qui serviront à un classifieur pour décider parmi 21 étiquettes de structure logique. Le même protocole que vu précédemment est choisi pour évaluer la création de groupe, à savoir un PMC comme classifieur et une sélection «au mieux par l'aléatoire». Le tableau 4.5 résume les résultats obtenus.

Nb. de variables	Méthode	
	Aléatoire	Partitionnement
56 (max)	100%	100%
35	86,9%	99,3%
25	65,0%	79,6%
15	51,8%	80,1%
10	35,1%	83,8%
5	17,9%	44,9%

TABLEAU 4.5 – Résultats normalisés de reconnaissance de structures logiques à partir de caractéristiques physiques en fonction de différentes tailles de sous-ensembles de variables

Tout comme les précédentes expérimentations, les conclusions sont identiques, la méthode de partitionnement donne toujours de meilleurs résultats que la méthode aléatoire. Même si les variables proviennent de caractéristiques de haut niveau, le comportement de la méthode proposée reste robuste. Il semblerait même que dans le cas présent, elle s'y prête parfaitement bien car en divisant la taille de l'entrée (déjà petite) par plus de cinq, on garde près de 84% de l'information d'origine. Notons que pour cet exemple (choisir 10 variables parmi 56), il faudrait tester près de 36 milliards d'ensembles pour trouver l'optimal avec une méthode exhaustive. Le tableau 4.6 présente le nom des observations physiques retenues pour un premier sous-ensemble de taille 10.

Position verticale	Largeur boîte	Espacement droite	Italique
Taille police	Indentation gauche	Encadré	Pourcentage numérique
Pourcentage ponctuation	Nombre de lignes		

TABLEAU 4.6 – Caractéristiques physiques choisies pour un premier sous-ensemble de taille dix

À titre d'information, le tableau 4.7 donne les résultats obtenus par une réduction (et non une sélection) de données effectuée par une ACP. On y voit qu'effectivement pour conserver plus d'information avec le moins de variables possible, il est préférable d'avoir recours à de la fusion plutôt qu'une sélection. Ces résultats donnent en quelque sorte une borne supérieure aux résultats que l'on pourrait trouver avec une méthode de sélection. On remarquera que les résultats sont très intéressants aussi bien sur la base MNIST où il y a à la fois un grand nombre de variables inutiles (comme les pixels des bords) et un fort lien entre les pixels (voisinage). La différence est moins contrastée sur les variables de haut niveau comme les indices physiques, bien qu'il existe des dépendances assez fortes entre certaines variables qui jouent en faveur d'une réduction.

En plus des résultats sur le taux d'information conservée, nous présentons dans le tableau 4.8 l'influence du choix de la réduction de valeurs propres. Comme développé en sous-section 4.4.4, nous avons expérimenté les cinq méthodes les plus connues pour déterminer la dimension

MNIST (7×7)		MNIST (28 × 28)		Indices physiques	
Nb. C.P.	Information	Nb. C.P.	Information	Nb. C.P.	Information
49 (max)	100%	784 (max)	99,8%	56 (max)	99,9%
35	100,6%	500	100%	35	99,9%
25	99,9%	150	100,1%	25	96,2%
15	97,2%	100	99,9%	15	93,0%
10	91,3%	50	99,9%	10	91,9%
5	74.21%	25	99,4%	5	83,7%

TABLEAU 4.7 – Information conservée par l'ACP en fonction de la base et du nombre de composantes principales retenu

k du sous-espace engendré par la matrice transposée des vecteurs propres.

Nb. caractéristiques	Nombre fixe		% fixe		% de variance		Kaiser (q=14)	Cattell (q=19)
	Nb.	Taux	%F	Taux	%V	Taux		
5	2	64,4%	2	61,6%	10	66,5%	69,2%	68,1%
	5	64,3%	5	72,1%	20	67,9%		
	10	60,3%	10	64,3	40	61,4%		
	15	59,2%	20	57,8%	60	63,6%		
10	2	78,4%		79,7%		81,1%	77,7%	82,3%
	5	79,8%		82,7%		73,5%		
	10	72,9%		77,1%		78,4%		
	15	70,0%		76,6%		72,6%		
20	2	85,4%		82,1%		82,3%	85,7%	86,1%
	5	84,9%		82,8%		86,1%		
	10	83,6%		83,3%		82,3%		
	15	82,6%		83,3%		78,8%		
30	2	85,2%		82,9%		84,2%	87,4%	88,0%
	5	86,8%		85,6%		85,8%		
	10	86,6%		86,5%		86,7%		
	15	86,3%		85,4%		87,7%		

TABLEAU 4.8 – Taux de reconnaissance de structures logiques en fonction de différentes tailles de sous-ensembles de variables et de méthodes de détermination de dimension k de l'espace réduit

Le choix de k a donc une influence certaine sur la qualité des sous-ensembles. Bien qu'un PMC soit un classifieur donnant d'assez bons résultats même pour un faible nombre de variables, on constate que si le partitionnement ne forme pas un «bon» premier groupe, les conséquences sur le taux de reconnaissance sont très visibles. Il semble que dans cet exemple, ainsi que pour d'autres effectués sur la base MNIST, le critère de Cattell sélectionnant $k = 19$ donne des résultats en moyenne supérieurs aux autres méthodes. Pour une méthode automatique, le critère de Cattell est plus performant que celui de Kaiser (en particulier pour un groupe de taille 10).

On remarquera qu'il utilise des vecteurs plus grands ($k = 19$), ce qui explique en partie ses bons résultats. D'un autre côté, le fait de choisir parfois une dimension plus petite (ici $k = 5$) semble donner des résultats très similaires mais cette constatation ne peut se faire qu'après avoir examiné un grand nombre de cas et nécessitera, comme évoqué en sous-section 4.4.4 une bonne connaissance des variables.

Nous avons fait aussi plusieurs tests sur la méthode permettant de clusteriser les lignes P_i de la matrice réduite des vecteurs propres U_k. Nous avons testé un algorithme de type carte auto-organisatrice et une méthode des centres mobiles *k-means* [MacQueen, 1967], qui tous deux donnent des résultats comparables. Parfois la carte SOM donnera de meilleurs résultats mais le nombre de paramètres à fixer empiriquement ajoute trop de biais à la méthode pour être robuste dans n'importe quelle situation. À paramètres fixes et à temps d'exécution semblable, un algorithme *k-means*, même non optimisé (Algo. 4) [Arthur et Vassilvitskii, 2007], donne des résultats plus convaincants en moyenne. Un seul paramètre, $iter_{max}$, est à fixer dans cette méthode. Pour stabiliser les centres des classes, sa valeur ne doit pas être nécessairement grande ; avec $iter_{max} = 2\,nb_{vec}$, on termine toujours sur un optimum local. Au pire des cas, la clusterisation est en $\mathcal{O}(n^3)$, n représentant le nombre total de variables.

Vec : les vecteurs à clusteriser
$centres$: matrice des centres des classes
$classes$: liste des affectations dans les classes des vecteurs à clusteriser
$somme$ et $centres$: initialisés aléatoirement avec des vecteurs d'entrée
$\forall i, somme[i][dim(Vec)] \leftarrow 1$
pour $iter$ *de* 1 *à* $iter_{max}$ **faire**
  pour i *de* 1 *à* nb_{vec} **faire**
    pour j *de* 1 *à* nb_{classe} **faire**
      $distances[j] \leftarrow dist_{eucl}(Vec[i] - centres[j])$
    fin
    $indice \leftarrow arg_j(min_j(distances[j]))$
    $classes[i] \leftarrow indice$
    $somme[indice][dim(Vec)] \leftarrow somme[indice][dim(Vec)] + 1$
    $somme[indice] \leftarrow somme[indice] + Vec[i]$
    $centres[indice] \leftarrow \frac{sommes[indice]}{sommes[indice][dim(Vec)]}$
  fin
fin

ALGORITHME 4 – Méthode de centres mobiles

Application du partitionnement au réseau de neurones perceptif

Nous avons testé une fois de plus le réseau de neurones perceptif sur la base d'articles scientifiques en suivant le même protocole qu'en section 3.3, page 67. Trois groupes de variables sont formés et au premier cycle perceptif, seul le groupe de plus petite taille est utilisé en entrée du réseau. Le tableau 4.9 donne un résumé des scores de reconnaissance obtenus avec l'introduction du partitionnement.

Classes	PMC	Réseau de neurones perceptif			
		Cycle 1	Cycle 2	Cycle 3	Cycle 4
Toutes	81,7%	45,2%	78,9%	90,2%	91,7%
La meilleure	98,9%	66,7%	85,3%	100,0%	100,0%
La plus mauvaise	0,0%	0,0%	0,0%	28,9%	28,9%
Facteur temps	1	0,7	1,45	1,85	2,1

TABLEAU 4.9 – Classification de structures logiques par un PMC et un RNP avec cycles perceptifs et partitionnement de l'espace d'entrée

Certaines formes, les plus simples, sont reconnues dès la première propagation en n'utilisant qu'un petit sous-ensemble des observations physiques. Inversement, certaines formes demandent l'ensemble complet des variables dès le cycle n°3 et, tout comme dans la version sans partitionnement, certaines d'entre elles ne sont toujours pas reconnues même après un cinquième ou un sixième cycle perceptif.

Les taux de reconnaissance sont assez similaires à ceux obtenus sans le partitionnement (Tab. 3.4, p. 68), la différence réside ici dans le temps mis à les obtenir ; au troisième cycle le score dépasse 90% avec seulement un allongement du temps par 1,85 contre 2,4 dans la précédente version. Au quatrième cycle, pour un temps de reconnaissance doublé, on obtient un taux de reconnaissance très proche de celui obtenu en triplant le temps avec la version sans cycles. Au cinquième cycle, le nouveau RNP atteint 92,2% avec un facteur de temps de 2,3 ce qui reste toujours inférieur au temps mis par l'ancien au quatrième cycle.

Les résultats qualitatifs sont eux aussi très similaires à ceux présentés au précédent chapitre, les formes non reconnues par l'un sont pratiquement les mêmes que celles non reconnues par l'autre, on a d'ailleurs une similarité de plus de 80% au cycle n°3 et plus de 90% au cycle n°4. Comme évoqué au cours de ce chapitre, le partitionnement joue juste un rôle d'accélérateur de la reconnaissance, il n'améliore pas la qualité de la reconnaissance. Il faut, dans notre cas, un cycle supplémentaire pour atteindre un score dépassant les 90% mais le RNP avec partitionnement l'obtient plus rapidement. On notera ici que nous utilisons toujours le même OCR commercial comme extracteur d'observations physiques et qu'il est difficile d'isoler le calcul de chaque variable, les facteurs de temps sont alors encore une fois de plus donnés au pire des cas. Il est bien sûr évident qu'avec une implémentation plus fine des outils d'extraction, les facteurs de temps auraient été plus tranchés.

Discussion

Nous avons formulé un certain nombre de remarques sur les paramètres à fixer dans la méthode de partitionnement en essayant d'analyser tous les cas défavorables pouvant se présenter lors de l'utilisation de l'algorithme. Bien qu'il soit utile de prévoir le comportement de l'algorithme en fonction de ses paramètres, nous avons toujours mené notre discussion comme s'il s'agissait d'une méthode de sélection optimale et non d'une méthode de partitionnement. Notre objectif est de créer des groupes de variables d'intérêt décroissant pour alimenter un réseau de neurones perceptif. Or, même si notre premier groupe de variables n'est pas l'optimal ou est moins performant qu'un groupe créé par une autre méthode que celle proposée, l'attitude du système

Classes	Nb. échantillons	Taux PMC	Taux RNP avec parti.
Titre Document	15	93,3%	100,0%
Auteur	44	88,6%	90,9%
Email	5	0,0%	80,0%
Adresse	21	47,6%	66,7%
Résumé	15	93,3%	100,0%
Mots-clés	14	92,8%	92,9%
Catégories	9	88,9%	100,0%
Introduction	73	80,8%	80,8%
Paragraphe	440	96,1%	95,7%
Section	92	97,8%	97,8%
Sous-Section	62	98,3%	98,4%
Sous-sous-section	17	76,4%	76,5%
Liste	69	97,1%	98,6%
Énumération	44	95,4%	97,7%
Flottant	105	91,4%	99,1%
Conclusion	38	28,9%	28,9%
Bibliographie	187	98,9%	98,9%
Algorithme	86	95,3%	97,7%
Copyright	9	88,8%	100,0%
Numéro page	30	96,6%	93,3%
Remerciements	10	70,0%	60,0%

TABLEAU 4.10 – Résultats détaillés du réseau de neurones perceptif et partitionnement au troisième cycle pour chaque classe

global de reconnaissance n'en sera pas plus modifiée car au final, pour des formes ambiguës, l'ensemble complet des variables sera sans doute utilisé après quelques cycles perceptifs. En effet, même si quelques variables auraient du être présentes dans le premier groupe, elles le seront sûrement dans le second groupe qui sera utilisé plus tard, en cas de problèmes. Quelque soit le partitionnement proposé, le RNP utilisera les variables dont il a besoin pour décider d'une forme. Si le partitionnement est de mauvaise qualité, il faudra simplement plus de cycles perceptifs pour certaines formes. La qualité sera la même, seule la réduction du temps de reconnaissance sera moins impressionnante.

La méthode que nous proposons comporte plusieurs avantages qui méritent d'être soulignés :

- c'est une méthode par filtre, elle se fait donc en prétraitement de la reconnaissance ;
- elle se fait indépendamment du classifieur ce qui permet aussi de construire des RNP en fonction des différents groupes de variables créés par le partitionnement ;
- elle repose sur des bases statistiques prouvées et chaque étape de l'algorithme converge vers un optimum ;
- aucun paramètre ne doit être fixé manuellement, la méthode est autonome si l'on utilise Cattell pour la réduction d'espace et un *k-means* pour la clusterisation ;

– la complexité globale est au pire des cas en $\mathcal{O}(n^3)$ où n est le nombre de variables (56 dans nos expérimentations) et permet d'obtenir un partitionnement en quelques secondes sur un ordinateur récent ;
– le partitionnement proposé peut aussi être utilisé comme fonction de score pour les variables. Il peut alors aider une autre méthode à trouver des groupes différents de variables en introduisant de nouveaux critères pour établir sa classification.

Ce dernier avantage est très important à souligner ; nous avons énoncé en début de chapitre qu'il était beaucoup plus facile de créer des groupes de variables en fonction du critère du temps d'extraction que du pouvoir informatif qui, lui, est difficile à caractériser. Grâce à la méthode de partitionnement, un score de « qualité » peut être affecté à une variable : les variables du premier groupe auront un score élevé, il sera plus faible pour celles du second et ainsi de suite jusqu'au dernier. Chaque variable possédant un poids, on peut utiliser un autre critère comme le temps d'extraction pour mettre à jour les scores et créer de nouveaux groupes. En fonction du but recherché (le pouvoir informatif en premier ou alors la rapidité), on accordera plus ou moins d'importance aux différents critères. On peut même extraire un autre critère du partitionnement à savoir la distance d'une variable avec le centre de son groupe. Trois critères (information, non-corrélation, temps d'extraction) seront alors disponibles pour qualifier les variables. Avec une fonction d'adéquation *(fitness function)* bien choisie, on peut alors facilement créer de nouveaux partitionnements en utilisant des méthodes d'optimisation comme par exemple les algorithmes génétiques [Sun et coll., 2004].

4.6 Conclusion

Le réseau de neurones perceptif est un système de reconnaissance efficace mais qui demande plusieurs phases d'extraction des entrées pour parfaire ses résultats. Nous avons montré que pour réduire le temps de reconnaissance, il n'était pas nécessaire de revoir le fonctionnement du réseau, mais de se concentrer sur l'obtention des données d'entrée dont le temps d'extraction est le vrai responsable de la limitation du système. Comme ces données nous sont fournies par des outils indépendants, nous n'avons que très peu de moyens pour les contrôler. Les propagations et les rétropropagations dans le RNP étant instantanées et n'ayant aucune influence sur les extractions des variables d'entrées, nous avons opté pour un partitionnement de l'espace d'entrée afin de réduire au maximum les extractions inutiles.

Nous avons proposé une amélioration du fonctionnement des cycles perceptifs en utilisant le partitionnement dont les groupes nous servent à alimenter le RNP. Plutôt que de fournir à chaque cycle l'ensemble complet des variables disponibles, le système débute toujours sa reconnaissance par un sous-ensemble de petite taille et, si les corrections sur cet ensemble ne sont pas suffisantes pour déterminer la forme, le système utilisera un autre sous-ensemble pour compléter le courant, trop restreint pour donner une étiquette. Si la forme est simple à reconnaître, peu de variables auront été extraites et inversement, seules les formes les plus difficiles nécessiteront éventuellement l'ensemble complet des variables.

Créer des groupes de variables peut se faire manuellement si l'on dispose de suffisamment de connaissances pour déterminer le partitionnement qui réduira le nombre de cycles perceptifs. Dans le cas contraire, si les données à manipuler sont trop complexes à analyser, le risque de créer des groupes non intéressants pour le classifieur est plus grand. La méthode que nous avons proposée est à mi-chemin entre sélection de variables et réduction de données. Nous avons montré que cette dernière fusionnait les variables d'origine et qu'il était alors impossible d'utiliser les cycles

perceptifs. D'un autre côté, les méthodes de sélection sont plus appropriées mais demandent des calculs coûteux et l'utilisation du classifieur. Le partitionnement proposé emploie les premières étapes de l'analyse en composantes principales comme de nombreuses autres méthodes par filtre, puis utilise les propriétés des composantes de la matrice des vecteurs propres pour regrouper les variables corrélées entre elles. En choisissant les variables dans chaque groupe, on crée de nouveaux ensembles qui forment le partitionnement. L'algorithme permettant de partitionner (ou de catégoriser les données) se fait en prétraitement de la reconnaissance, sa complexité est cubique en la dimension des vecteurs d'entrée et ne nécessite aucun autre paramètre.

Le partitionnement renforce l'aspect perceptif du RNP, car en plus de la décomposition par palier d'interprétation du processus de reconnaissance, il propose une hiérarchisation de l'intégration des observations physiques. Le système est complètement adaptatif, même une fois entraîné, ce qui renforce la mixité entre l'approche par les données et l'approche par le modèle car, dans notre cas expérimental, la connaissance du modèle de la structure physique et logique peut aussi aider à créer les groupes d'observations physiques. Comme dans le cas de la perception humaine, le RNP est capable d'adapter la quantité de travail nécessaire en fonction de la difficulté de la forme à reconnaître.

Le réseau de neurones perceptif muni de sa méthode de partitionnement est un outil possédant de nombreux avantages pour des problèmes de reconnaissance de formes structurées et nécessitant des extractions d'indices de haut niveau. La reconnaissance, dynamique, se fait par plusieurs allers-retours entre les entrées et les sorties. Ce mécanisme fonctionne mais il peut être amélioré. Nous allons détailler dans le prochain chapitre comment il est envisageable de modifier l'algorithme d'apprentissage afin que le réseau puisse, lors de la reconnaissance, s'adapter aux variations des données d'origine dues aux corrections faites à chaque cycle. Après une brève introduction aux réseaux de neurones dynamiques, nous montrerons comment adapter un réseau de neurones à décalage temporel pour adapter notre système aux données lors des corrections des variables d'entrée.

Chapitre 5

Réseau de neurones dynamique perceptif

Reconnaître une forme avec le réseau de neurones perceptif demande d'effectuer plusieurs cycles entre les données et leur interprétation. Les corrections, nécessaires à l'affinement de la solution, qui en résultent impliquent une modification des valeurs des entrées. Le réseau est appris une seule fois avec une base de formes fixes, bien qu'au cours de la reconnaissance, un vecteur de caractéristiques différent soit utilisé pour décider de la même forme. Nous allons montrer dans ce chapitre comment intégrer cette dynamique durant l'apprentissage afin de proposer des réponses plus adéquates après chaque cycle perceptif. Après un bref aperçu des réseaux de neurones dynamiques, nous analyserons une solution basée sur un réseau de neurones à décalage temporel pour résoudre le problème de la variation des données au cours du temps.

Sommaire

5.1 Réseau de neurones perceptif et correction des entrées 103
5.2 Réseaux dynamiques 104
5.2.1 Réseaux statiques récurrents 104
5.2.2 Autres réseaux dynamiques 106
5.2.3 Difficultés des réseaux dynamiques 107
5.2.4 Choix du réseau 107
5.3 Réseau à décalage temporel 108
5.3.1 Topologie et fonction d'activation 108
5.3.2 Apprentissage 109
5.4 Réseau de neurones dynamique perceptif 112
5.5 Expérimentations 115
5.6 Perspectives 117
5.7 Conclusion 121

5.1 Réseau de neurones perceptif et correction des entrées

Nous avons décrit au cours du chapitre 3 le fonctionnement du réseau de neurones perceptif, de la création de la topologie jusqu'à la reconnaissance. Nous avons particulièrement mis en avant l'atout majeur de ce réseau à savoir la correction des entrées. Peu de systèmes dans la

littérature reviennent sur leurs entrées une fois la reconnaissance effectuée. Ceci reste vrai dans le cadre de l'analyse de structures logiques de documents et particulièrement lorsque des méthodes dirigées par les données sont employées. Il existe certes des systèmes ayant recours au rejet ou à une post-correction, mais peu agissent comme le RNP mettant en doute la qualité de ses entrées et faisant intervenir l'information de contexte pour corriger les entrées fautives. Le RNP « adapte » donc la forme pour qu'elle puisse être mieux reconnue au prochain cycle perceptif. L'adaptation est, dans notre cas, une correction de la segmentation qui implique nécessairement une modification d'un certain nombre d'autres variables du vecteur d'entrée.

La question que nous allons développer dans les prochaines sections est de savoir comment tenir compte de ces entrées changeantes avec le même réseau. Jusqu'à maintenant le RNP était basé sur un modèle statique, c'est-à-dire que le temps n'influe pas sur les entrées et il était entraîné sur une base fixe, celle correspondant au dernier cycle perceptif. L'idée serait d'exploiter l'information de correction non seulement pendant la reconnaissance mais aussi pendant l'apprentissage afin de procurer des réponses plus adaptées en fonction de l'avancement des cycles perceptifs. Les formes x_i sont en effet différentes à chaque cycle (sinon elles sont déjà classifiées), x_i devrait être plutôt nommé $x_i(t)$ où t est le numéro du cycle perceptif courant. Comme les poids du réseau $w_{l,i,j}$ sont constants et déterminés pour des $x_i(t = \infty)$, ils ne sont donc pas appris de manière optimale pour des données $x_i(t)$ avec $x_i(0) \neq \ldots \neq x_i(n)$. L'idéal serait de construire un réseau f avec des poids $w_{l_i,j}(t)$ de telle sorte que $t_1 \neq t_2 \Rightarrow f(x(t_1), t_1) \neq f(x(t_2), t_2)$ même si $x(t_1) = x(t_2)$.

Plusieurs réseaux de neurones artificiels sont capables de traiter des informations temporelles. Nous allons nous servir de l'un d'eux, le réseau à décalage temporel, et nous justifierons son choix après avoir étudié sont fonctionnement et décrit son apprentissage. Avant de nous intéresser à ce réseau qui sera utilisé dans le RNP, nous donnerons un rapide aperçu des différentes architectures possibles. Bien que nous ne les employions pas dans notre méthode, nous mettrons un peu plus en avant les réseaux statiques récurrents qui permettent d'avoir un exemple plus détaillé du fonctionnement de réseaux dynamiques et nous étudierons plus particulièrement comment réaliser un apprentissage pour ce type d'architecture. Pour avoir des explications complémentaires, on pourra se référer à des articles tels que [Pearlmutter, 1995 ; Baldi, 1995] qui donnent des indications plus détaillées sur les réseaux qui ne seront pas aussi développés dans ce mémoire que le réseau statique récurrent et le réseau à décalage temporel.

5.2 Réseaux dynamiques

Les principaux réseaux supervisés présentés jusqu'à maintenant étaient des réseaux statiques non bouclés. Si l'on reprend le Perceptron multicouche, on s'aperçoit qu'il n'est en fait qu'une approximation grossière de la réalité ; si l'on se réfère au fonctionnement du cerveau humain, on constate que le cortex est divisé en plusieurs couches composées d'un nombre important de connexions. Les interactions au sein d'une même couche sont évidemment très nombreuses auxquelles se rajoutent les connexions entre les couches qui elles sont présentes dans le PMC. Il est plus commode d'omettre le bouclage au sein des couches de sorte que chaque neurone ne reçoive que les signaux de la couche précédente.

5.2.1 Réseaux statiques récurrents

Il existe cependant des réseaux permettant de tenir compte de toutes les interactions possibles entre les neurones que l'on appelle réseaux statiques récurrents. Dans ce type de réseau, le

terme récurrent fait référence à la topologie et non à l'introduction du temps dans le calcul des activations. On appelle aussi ce type d'architecture réseaux bouclés car les neurones ne sont pas forcément organisés en couches successives. On peut imaginer des connections supplémentaires partant de n'importe quel neurone et pouvant être reliées à n'importe quel autre. La figure 5.1 montre un exemple extrême d'architecture totalement bouclée.

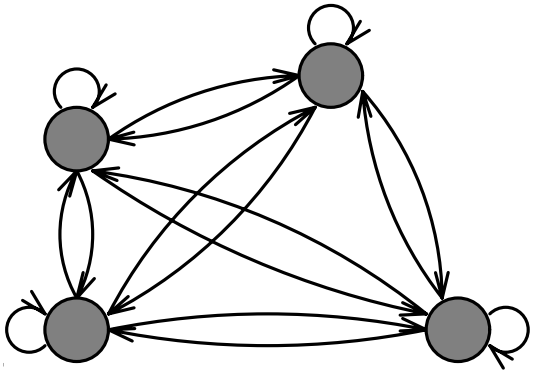


FIGURE 5.1 – Réseau à quatre neurones complètement bouclé

L'introduction du bouclage à n'importe quel niveau rend les formules vues jusqu'à maintenant inadaptées. Le point d'équilibre d'un tel réseau est plus difficile à obtenir mais il est tout de même démontré qu'il peut être obtenu lorsque les poids des connexions entre deux neurones distincts sont identiques ($\forall(i,j), w_{ij} = w_{ji}$).

Si on note V la réponse du réseau, pour chaque neurone i parmi les N possibles, il faut vérifier :

$$V_i = f(s_i) = f\left(\sum_{j=1}^{N} w_{ij} V_j\right) \tag{5.1}$$

Les travaux de [Pineda, 1987 ; Pineda, 1988 ; Pineda, 1989], [Almeida, 1987] et [Rohwer et Forrest, 1987] montrent conjointement que, sous certaines conditions, la rétropropagation du gradient peut être étendue à ce modèle. L'algorithme modifié s'appelle rétropropagation récurrente. Il doit simplement vérifier que les entrées et les sorties sont fixes et connues, les entrées ne devant recevoir aucun signal des autres neurones. En dérivant V_i suivant le temps dans le cas de la relaxation d'un système dynamique, on obtient :

$$\tau_i \frac{\partial V_i}{\partial t} = -V_i + f\left(\sum_{j=1}^{N} w_{ji} V_j\right) + x_i \tag{5.2}$$

Il n'est néanmoins pas assuré que cette équation converge vers un point d'équilibre. Il est tout à fait possible d'être confronté à un mouvement oscillatoire voire à un comportement chaotique [Hertz et coll., 1991], il faut que les τ_i soient convenablement choisis pour amener le système vers un état stable.

Si on admet que l'équation converge vers un point d'équilibre, il existera un i tel que $\frac{\partial V_i}{\partial t} = 0$ et le réseau vérifiera la propriété :

$$V_i = f\left(\sum_{j=1}^{N} w_{ji} V_j\right) + x_i \tag{5.3}$$

Il faut alors minimiser l'erreur quadratique $E(w) = \sum_i^N E_i^2$ avec toutefois $E_j = d_j - V_j$ pour un neurone de sortie et 0 pour tous les autres neurones. Si une descente de gradient est utilisée comme dans le cas statique, on obtient comme variation des poids :

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} = -\eta \frac{1}{2} \sum_k \frac{\partial E_k^2}{\partial w_{ij}} \tag{5.4}$$

$$= -\eta \sum_k E_k \frac{\partial E_k}{\partial w_{ij}} \tag{5.5}$$

Or $E_k = d_k - V_k$ donc :

$$\Delta w_{ij} = \eta \sum_k E_k \frac{\partial V_k}{\partial w_{ij}} \tag{5.6}$$

En développant le terme $\frac{\partial V_k}{\partial w_{ij}}$ et en effectuant deux inversions (Annexe D, p. 149), la rétropropagation se résume à relaxer le réseau original jusqu'à un point fixe, comparer la sortie du réseau aux valeurs désirées de façon à obtenir les erreurs qui servent à alimenter le réseau adjoint, relaxer ce dernier afin d'avoir les Y_q et utiliser $\Delta w_{ij} = \eta \sum_k E_k (L^{-1})_{kj} f'(s_j) V_i$ pour trouver les poids d'origine.

5.2.2 Autres réseaux dynamiques

Il existe plusieurs autres classes de réseaux de neurones qui se détachent du simple Perceptron multicouche. Nous présenterons dans la section suivante le réseau à décalage temporel, utilisant un PMC et une ligne de temporisation pour prédire des séries temporelles. Les réseaux à retour de contexte *(networks with feedback dynamics)* ont des liens partant en sens inverse dans les couches contrairement aux réseaux simplement *feedforward*. Ils sont aussi appelés réseaux récurrents et leur apprentissage nécessite d'ailleurs une adaptation récursive de l'algorithme de rétropropagation. Des architectures, à mi-chemin entre le réseau à décalage temporel et le réseau à retour de contexte, utilisent le résultat des sorties comme entrée *(networks with output feedback)*. On peut d'ailleurs avoir recours à une seconde ligne de temporisation pour apprendre facilement ce type de réseau sur la base d'un PMC [Narendra et Parthasarathy, 1990]. Les réseaux à retour d'état *(networks with state feedback)* ont une topologie encore plus générale dans le sens où le réseau n'est pas forcément organisé en couches, chaque neurone est interconnecté avec les autres et participe au vecteur d'état avec aussi des connexions *feedback*. Le réseau temps continu de Hopfield *(continuous-time Hopfield net)* fait intervenir la composante temporelle dans un réseau à une couche complètement connecté [Hopfield, 1982]. Une version discrète *(discrete-time Hopfield network)*, ayant les comportements proches de la version continue, propose des équations approchées du cas continu et un remplacement de la fonction sigmoïde par une fonction seuil. Les réseaux récurrents à temps continu

(continuous-time recurrent neural networks) [Pineda, 1988] et leur simplification à temps discret *(discrete-time recurrent neural networks)* [Williams et Zipser, 1989] proposent des équations différentes des réseaux de Hopfield mais on peut les mettre en correspondance avec une transformation affine. Les versions discrètes, ayant des capacités proches de versions continues sont souvent privilégiées car l'apprentissage est plus facile à réaliser ; il est même possible de « déplier » le réseau à travers le temps et d'utiliser une technique classique de rétropropagation sur un réseau *feedforward* appelée rétropropagation dans le temps *(backpropagation trough time)* ou d'utiliser la propagation récurrente comme dans la sous-section précédente.

5.2.3 Difficultés des réseaux dynamiques

Les réseaux dynamiques sont classiquement appris soit avec la rétropropagation dans le temps (BPTT) soit avec la rétropropagation récurrente (RTRL). D'un point de vue complexité calculatoire, la première est en $\mathcal{O}(n^2 d)$ avec n le nombre de poids et d la longueur de la séquence, la seconde est en $\mathcal{O}(n^4 d)$. Sur le plan de la complexité spatiale, les complexités sont respectivement en $\mathcal{O}(nd)$ et $\mathcal{O}(n^3)$. La BPTT est donc plus performante si elle doit reconnaître des séquences courtes. Si on analyse plus finement [Logar et coll., 1993], la BPTT demande $7n^2d + 71nd + 2n^2$ opérations contre $t(n^4 + 5n^3 + 6n^2 + 30n + 2n^3m + 6nm)$ où m est la taille de l'entrée, et d'après leurs tests sur des cas réels où n est petit devant t, il semblerait que les temps relatifs soient favorables cette fois-ci pour RTRL, les deux restant toutefois bien plus lents qu'un PMC statique.

Outre la complexité calculatoire, les problèmes de convergence des réseaux dynamiques sont plus nombreux que dans les cas statiques. Les problèmes de minima locaux correspondant à une erreur élevée sont plus fréquents [Szilas, 1995]. Le pas d'apprentissage est aussi beaucoup plus difficile à fixer : dans les algorithmes d'apprentissage, la convergence n'est assurée que pour un pas infiniment petit. Dans des cas pratiques, ceci ne peut pas être réalisable et le nombre d'époques doit alors être important. La surface d'erreur contient aussi plus de « plateaux » sur lesquels l'algorithme n'évolue pas, pensant avoir atteint un minimum local. Ces problèmes de «plateaux» sont amplifiés lors d'une implémentation informatique car, à ces endroits, le gradient de l'erreur est proche de zéro ce qui peut provoquer de graves instabilités numériques. Tous ces problèmes sont amplifiés lorsque la séquence à étudier est longue ; plus la simulation se prolonge, plus le problème de l'explosion du gradient est probable (l'erreur instantanée sur un poids du réseau croît très vite). Des exemples théoriques montrent aussi clairement que le gradient peut osciller avec une amplitude de plus en plus élevée sur des fonctions périodiques comme par exemple $t \mapsto sin(\omega t)$.

5.2.4 Choix du réseau

Nous avons focalisé notre discussion sur les réseaux récurrents qui ne sont pas spécialement adaptés au problème que nous avons soulevé au début du chapitre. Leur étude se révèle néanmoins intéressante car, que ce soit du point de vue fonctionnel ou des difficultés algorithmiques et techniques à résoudre, des problèmes similaires vont se retrouver dans les réseaux temporels mais non récurrents. De plus, nous proposerons une perspective d'évolution de notre réseau vers une version *feedback*, qui reprendra les principes vus jusqu'à maintenant.

Comme évoqué dans la dernière sous-section, d'autres réseaux dynamiques sont possibles pour résoudre le problème que nous nous sommes fixé. De part l'architecture actuelle du réseau de neurones perceptif, notre choix doit se porter a priori sur une architecture à couches. Les réseaux complètement connectés ne sont donc pas a priori nécessaires pour notre système. Nous avons besoin de retenir en mémoire les différentes variations des états d'une entrée pour que le réseau fournisse une réponse adéquate. Celui permettant de traiter des séquences temporelles et ayant une architecture et un fonctionnement proche du Perceptron multicouche est le réseau à décalage temporel. Nous avons déjà évoqué son utilisation dans le chapitre 2 en analyse et reconnaissance de documents, bien qu'il soit encore sous-représenté. Après la description de sa topologie et de son calcul d'activation tenant compte de la dimension temporelle, nous montrerons comment l'algorithme de rétropropagation du PMC peut être étendu pour la détermination des poids. Après un bref aperçu des possibilités théoriques et appliquées ce type de système nous présenterons comment l'adapter au réseau de neurones perceptif.

5.3 Réseau à décalage temporel

5.3.1 Topologie et fonction d'activation

Nous allons nous intéresser à une classe particulière d'architecture de réseaux qui sont appelés réseaux à décalage temporel *(time-delay neural networks)* [Lang et coll., 1990] aussi connus sous le nom de réseaux non bouclés à réponse impulsionnelle finie *(finite impulse neural networks)* [Wan, 1994 ; Wan, 1993]. Ce type de réseau a été très peu utilisé dans l'analyse de documents bien qu'il ait été appliqué avec succès dans les domaines de la reconnaissance de la parole et la prédiction de séries temporelles.

Le réseau à décalage temporel (RDT) est similaire à un Perceptron multicouche dans son aspect propagation directe *(feedforward)*. Sa différence tient au fait que ses entrées peuvent dépendre de sorties d'autres neurones non seulement au temps présent t mais aussi durant un nombre D d'étapes antérieures $(t-1, t-2, \ldots, t-D)$. La sortie d'un neurone i à l'instant t est donnée par :

$$s_i(t) = f\left(\sum_{j=1}^{i-1}\sum_{k=0}^{D} x_j(t-k)w_{ij}(k)\right) \tag{5.7}$$

Nous nous intéresserons plus particulièrement à une restriction du cas général qui consiste à n'utiliser des délais que sur les neurones d'entrée. Il porte le nom de réseau à entrées retardées *(input delayed neural network)*. S'il est évident que les fonctions calculables par ce dernier le sont aussi par un RDT, la réciproque est vraie également [Wan, 1994].

La formule présentée précédemment a comme inspiration biologique la modélisation plus fine de l'état d'un neurone suite à une excitation ; le processus décrivant la transmission des signaux entre les neurones est essentiellement linéaire. L'activation, ou plus précisément le potentiel cellulaire, s'écrit en conséquence :

$$s_i(t) = \sum_j \int_0^t w_{i,j}(\tau)x_j(t-\tau)d\tau \tag{5.8}$$

la dimension temporelle simulant ici la dissipation des signaux émis par les neurones à travers le temps.

Comme dans le cas statique, la sortie du neurone dépend des neurones incidents j. La différence se fait au niveau de l'introduction de la dimension temporelle faisant intervenir des résultats provenant d'époques antérieures. Cette équation se transpose facilement dans le cas discret en la formule :

$$s_i(n) = \sum_j \sum_{k=0}^{D} w_{j,i}(k) x_j(n-k) \tag{5.9}$$

En utilisant des notations vectorielles des poids et des entrées :

$$W_{i,j} = [w_{i,j}(0), \ldots, w_{i,j}(D)]^T \text{ et } X_i(n) = [x_i(n), x_i(n-1), \ldots, x_i(n-D)]^T \tag{5.10}$$

il vient :

$$s_i(n) = \sum_j W_{j,i} \cdot X_j(n) \tag{5.11}$$

La constante D, ou ordre des connexions, peut être prise comme variable globale du réseau, sans perte de généralité. Elle est la « mémoire » du réseau. Le symbole $\cdot$ signifie dans le cas présent le produit scalaire et non pas la multiplication. En raison de l'absence de récurrence (S.-Sec. 5.2.1), l'ensemble du réseau demeure à réponse impulsionnelle finie, il reste dans un certain sens très proche du cas statique comme nous allons le montrer dans la sous-section suivante.

5.3.2 Apprentissage

Notation

Les notations utilisées sont similaires à celles du cas statique vu au chapitre 2. Nous introduisons la composante temporelle ainsi que la notation vectorielle pour simplifier les écritures par la suite :

- L le nombre de couches ;
- $w^l_{i,j}(t)$ le poids à l'instant t entre le neurone i de la couche $l-1$ au neurone j de la couche l ;
- $W^l_{ij} = [w^l_{ij}(0), w^l_{ij}(1), \ldots, w^l_{ij}(D)]^T$;
- $a^l_j(t)$ la sortie à l'instant t du neurone j de la couche l ;
- $A^l_i(t) = [a^l_i(t), a^l_i(t-1), \ldots, a^l_i(t-D)]$;
- $s^l_j(t)$ l'entrée du neurone j de la couche l à l'instant t.

Avec $s^l_j(t) = \sum_i W^l_{ij} \cdot A^{l-1}_i$ et $a^l_j = f(s^l_j(t))$ où f est une fonction d'activation comme la sigmoïde et comme cas particuliers : $x_i(t) = a^0_i(t)$ la $i^{\text{ème}}$ entrée et $y_i(t) = a^L_i(t)$ la $i^{\text{ème}}$ sortie.

Rétropropagation temporelle

Dans ce type d'architecture, la dimension temporelle s'introduit directement dans le calcul de l'erreur entre sortie attendue et sortie calculée. À l'instant t on a :

$$E(t) = D(t) - Y(t) \tag{5.12}$$

En reprenant la formulation du réseau statique, le calcul de l'erreur totale quadratique sur des données de longueur T s'écrit :

$$Err = \sum_{k=1}^{T} E(k)^T E(k) \tag{5.13}$$

L'une des solutions envisageables pour résoudre la minimisation de Err est de se rapporter à un cas identique à celui du réseau statique. Pour s'affranchir de la composante k, il est possible de déplier le réseau suivant le temps en plusieurs réseaux statiques indépendants de la variable k. Les inconvénients de ce procédé sont d'une part la complexité en taille du réseau déplié, d'autre part, le fait que le calcul du gradient doit se faire individuellement puis être combiné pour garder la même valeur finale quel que soit l'instant.

Une autre façon de procéder consiste à dériver tout de même l'erreur :

$$\frac{\partial Err}{\partial W_{ij}^l} = \sum_{k=1}^{T} \frac{\partial E(k)^{\mathrm{T}} E(k)}{\partial W_{ij}^l} \tag{5.14}$$

avec la supposition que W_{ij}^l soit constant. Comme la variation des poids accumulés sur une période T est faible, ce choix constitue une bonne approximation (sous réserve d'utiliser un petit pas d'apprentissage). On peut introduire dans le calcul de la dérivée le terme $s_j^l(t)$, ce qui donne :

$$\frac{\partial Err}{\partial W_{ij}^l} = \sum_{k=1}^{T} \frac{\partial Err}{\partial s_j^l(k)} \frac{\partial s_j^l(k)}{\partial W_{ij}^l} \tag{5.15}$$

En remplaçant $s_j^l(k)$ par sa valeur on a :

$$\frac{\partial s_j^l(t)}{\partial W_{ij}^l} = \frac{\partial (W_{ij}^l \cdot A_i^{l-1}(t))}{\partial W_{ij}^l} = A_i^{l-1}(t) \tag{5.16}$$

Pour le terme $\frac{\partial Err}{\partial s_j^l(t)}$, on pose :

$$\delta_j^l(t) = \frac{\partial Err}{\partial s_j^l(t)} \tag{5.17}$$

De fait, évaluer $\frac{\partial Err}{\partial W_{ij}^l}$ revient à utiliser :

$$\Delta W_{ij}^l(t) = -\eta \delta_j^l(t) A_i^{l-1}(t) \tag{5.18}$$

Comme dans le cas des réseaux statiques, seul le terme $\delta_j^l(t)$ reste à calculer. Il suffit d'utiliser la rétropropagation de l'erreur en partant de la couche de sortie pour évaluer le terme $\delta_j^l(t)$, puis de redescendre jusqu'aux entrées.

Pour une couche de sortie, en développant le terme $\delta_j^l(t)$ pour $l = L$, on trouve :

$$\delta_j^L(t) = -2 e_j(t) f'(s_j^L(t)) \tag{5.19}$$

Pour une couche quelconque, $s_j^l(t)$ n'influence plus uniquement l'erreur $e_j(t)$ mais les $s_j^{l+1}(t)$ de la couche suivante :

$$\delta_j^l(t) = \sum_m \sum_k \frac{\partial Err}{\partial s_m^{l+1}(k)} \partial s_m^{l+1}(k s_j^l(t) \tag{5.20}$$

Si l'on poursuit une dérivation complète de tous les termes individuels (Annexes de [Wan, 1994]), l'algorithme se résume de la manière suivante :

$$\Delta W_{ij}^l = -\eta \delta_j^{l+1}(t) \cdot A_i^{l-1}(t) \tag{5.21}$$

$$\delta_j^l(t) = \begin{cases} -2e_j(t) \cdot f'(s_j^L(t)) & \text{si} \quad l = L \\ f'(s_j^l(t)) \cdot \sum_k \Phi_k^{l+1}(t)^T W_{jk}^{l+1} & \text{si} \quad l < L \end{cases} \tag{5.22}$$

$$\Phi_k^l(t) = [\delta_k^l(t), \delta_k^l(t+1), \ldots, \delta_k^l(t+D)]^T \tag{5.23}$$

On retrouve pratiquement la formule classique de rétropropagation du gradient pour les réseaux statiques si l'on remplace les termes vectoriels A, W et Φ par leurs équivalents scalaires. La différence se fait au niveau du calcul de δ que l'on obtient par filtrage arrière dans les neurones.

Applications du réseau à décalage temporel

Nous avons vu qu'il existait en section 5.2 des réseaux purement dynamiques, utilisant des équations différentielles dans les équations régissant le fonctionnement des neurones. Le réseau à décalage temporel, bien qu'il puisse être lui aussi utilisé pour prédire des séries temporelles, reste tout de même une conversion d'un réseau statique à la composante temporelle. Il n'est finalement qu'un PMC qui convertit une séquence temporelle en une forme statique en dépliant la séquence à travers le temps (sur une période finie D).

On peut voir le RDT comme un PMC qui serait alimenté par des séquences d'entrée passant par une ligne de temporisation *(tapped delay line)* parfois appelée registre à décalage. Elle agit à la manière d'une vis sans fin qui décale dans le temps les différents états de l'entrée (Fig. 5.2).

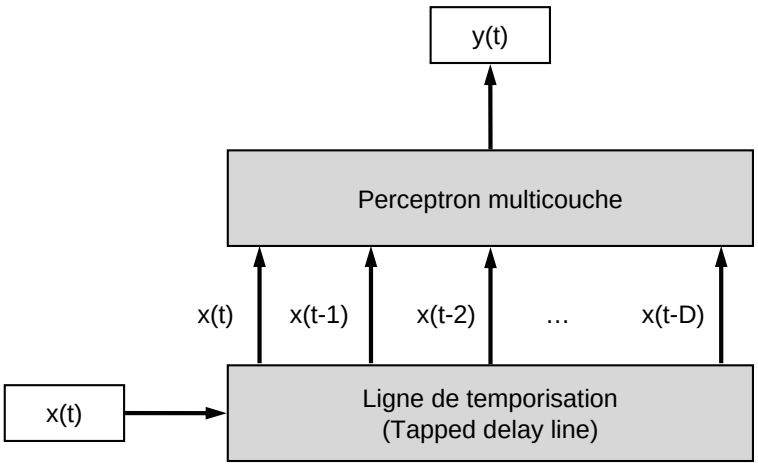


FIGURE 5.2 – Vue schématique d'un réseau à décalage temporel

Il est capable de modéliser des systèmes où les sorties ont une dépendance temporelle finie des entrées : $y(t) = F(x(t), x(t-1), \ldots, x(t-D))$. Quand F est une combinaison linéaire, l'architecture du RDT est équivalente à un filtre linéaire à réponse impulsionnelle finie *(linear FIR filter)*. Il est même capable de prédire des séquences chaotiques [Hush et Horne, 1993]. Il a été utilisé avec succès dans les domaines de la parole [Sejnowski et Rosenberg, 1987 ; Lang et coll., 1990 ; Sugiyama et coll., 1991], de la prédiction de trafic routier [Zhong et coll., 2006], la prédiction de séries temporelles non linéaires [Lapedes et Farber, 1987], la génération de trajectoires [Simard et Le Cun, 1992], dans la modélisation de phénomènes physiques complexes [Marques et coll., 2005], ou bien encore dans la reconnaissance des formes [Wöhler et

Anlauf, 1999] ou de signatures [Bromley et coll., 1994]. Il a même été étendu au réseau à fonction de base radiale [Moody et Darken, 1989].

Les RDT sont encore assez peu employés dans l'analyse logique de documents, bien que l'on dénote un intérêt croissant de leur utilisation [Mirowski et coll., 2007]. Ils sont généralement utilisés pour des tâches relatives au traitement de l'image ou à la reconnaissance du texte. On retrouvera des références pour les problèmes de reconnaissance d'écriture en ligne ou hors ligne. Dans le cas hors ligne, les auteurs considèrent généralement que l'axe horizontal de l'image du mot représente la composante temporelle. Les RDT sont capables de donner à la fois la classification de la forme et l'information de point de coupure dans le cas de la reconnaissance de mots. Dans [Martin, 1993], c'est une fenêtre se déplaçant horizontalement qui alimente les entrées du RDT. Le réseau est entraîné à reconnaître un caractère bien centré dans la fenêtre et à donner aussi son étiquette dans le cas d'un bon placement. Les RDT sont aussi employés pour reconnaître des caractères isolés comme l'ont fait [Pfister et coll., 2000] dans le cadre de l'analyse de codes postaux. Un RDT sert de classifieur principal pour les chiffres déformés par homothétie. Pour confirmer le code postal, un HMM est utilisé pour reconnaître le nom des villes. Les auteurs précisent qu'ils obtiennent plus de 99% de taux de reconnaissance sur la base NIST. Dans le cas de l'écriture en ligne, [Schenkel et coll., 1994] utilisent un RDT pour estimer les probabilités a posteriori des caractères dans un mot. Un HMM segmente ensuite le mot en caractères. Le système a été entraîné sur plus de 26 000 mots cursifs et testé sur plus de 600 nouveaux mots. Avec l'aide d'un dictionnaire, ils obtiennent environ 80% de bonne reconnaissance sur les mots longs et 70% sur les mots courts.

5.4 Réseau de neurones dynamique perceptif

La section précédente a montré le fonctionnement et l'intérêt d'un réseau à décalage temporel. S'il trouve parfois des applications en analyse et reconnaissance de documents, on le retrouvera majoritairement dans le cas de la reconnaissance de l'écriture. Il n'a jamais été appliqué, à notre connaissance, sur l'étiquetage ou l'analyse de la structure logique. Les méthodes dirigées par le modèle sont largement dominantes pour qu'un réseau dynamique, temporel de surcroît, soit utilisé dans la reconnaissance de formes qui n'ont pas de dimension temporelle.

Le problème que nous nous sommes posé est de savoir comment intégrer la variation des entrées $x(t)$ après chaque cycle perceptif. La solution du RDT semble être une bonne réponse car c'est un réseau qui permet de tenir compte d'une série d'observations et de procurer une réponse en adéquation avec cette série. Si l'on observe les expérimentations (Sec. 3.3, p. 67), on s'aperçoit au final que peu de cycles perceptifs sont nécessaires à l'obtention d'une réponse correcte. La séquence à observer devrait donc être courte. Les remarques négatives faites à l'encontre du RDT (S.-Sec. 5.2.3), sont alors nettement moins importantes car D est petit. On retrouve la complexité théorique d'un PMC même si dans la pratique les temps sont évidemment allongés.

L'utilisation du RDT ne pose donc a priori aucun inconvénient calculatoire et les problèmes de convergence devraient être minimes (toujours sous la réserve de D petit). De plus, lorsque les $x(t)$ vont varier au cours des cycles perceptifs, la différence entre deux vecteurs successifs ne devrait pas être importante. Si l'on considère que les cycles ne corrigent que la segmentation, certaines composantes du vecteur d'entrée ont de grandes chances de rester totalement identiques surtout si la boîte englobante est peu modifiée. Si les cycles ne corrigent pas la segmentation mais relancent, par exemple, certains outils d'extraction pour donner un résultat plus fiable

moyennant un temps de calcul plus long, on peut alors estimer que $\|x(t+1)-x(t)\| < \varepsilon(t)$, avec $\varepsilon(t)$ faible et de plus décroissant au cours du temps. Au pire des cas, il devrait toujours y avoir assez de composantes similaires entre deux entrées consécutives ($card\{i, |x_i(t+1)-x_i(t)| < \eta\}$ grand par rapport à la dimension de x). Le RDT est de toute manière capable d'absorber de grandes variations, comme évoqué dans la précédente section, il peut même prédire des séries chaotiques. La «continuité» des données fournies au réseau permet d'avoir une meilleure stabilité du système ainsi qu'un apprentissage lui aussi stable et plus rapide.

La nouvelle topologie du RNP, que nous appellerons à présent réseau de neurones dynamique et perceptif (RNDP), est similaire à l'ancienne car la ligne de temporisation permet de faire le passage des données temporelles au réseau avec l'ancienne structure (Fig. 5.3). La ligne de temporisation permet de conserver tous les concepts du RNP, les neurones représentent toujours les mêmes concepts et sont toujours organisés en couches. Dans l'implémentation, les connexions ne sont plus des scalaires mais des vecteurs ; plus précisément, les valeurs des poids sont des vecteurs, les liens entre les neurones restent inchangés. L'activation des neurones est elle aussi un vecteur ayant la même dimension que la ligne de temporisation. On ne retiendra lors de la reconnaissance que la dernière valeur de sortie, à l'instant présent, pour assigner les étiquettes logiques.

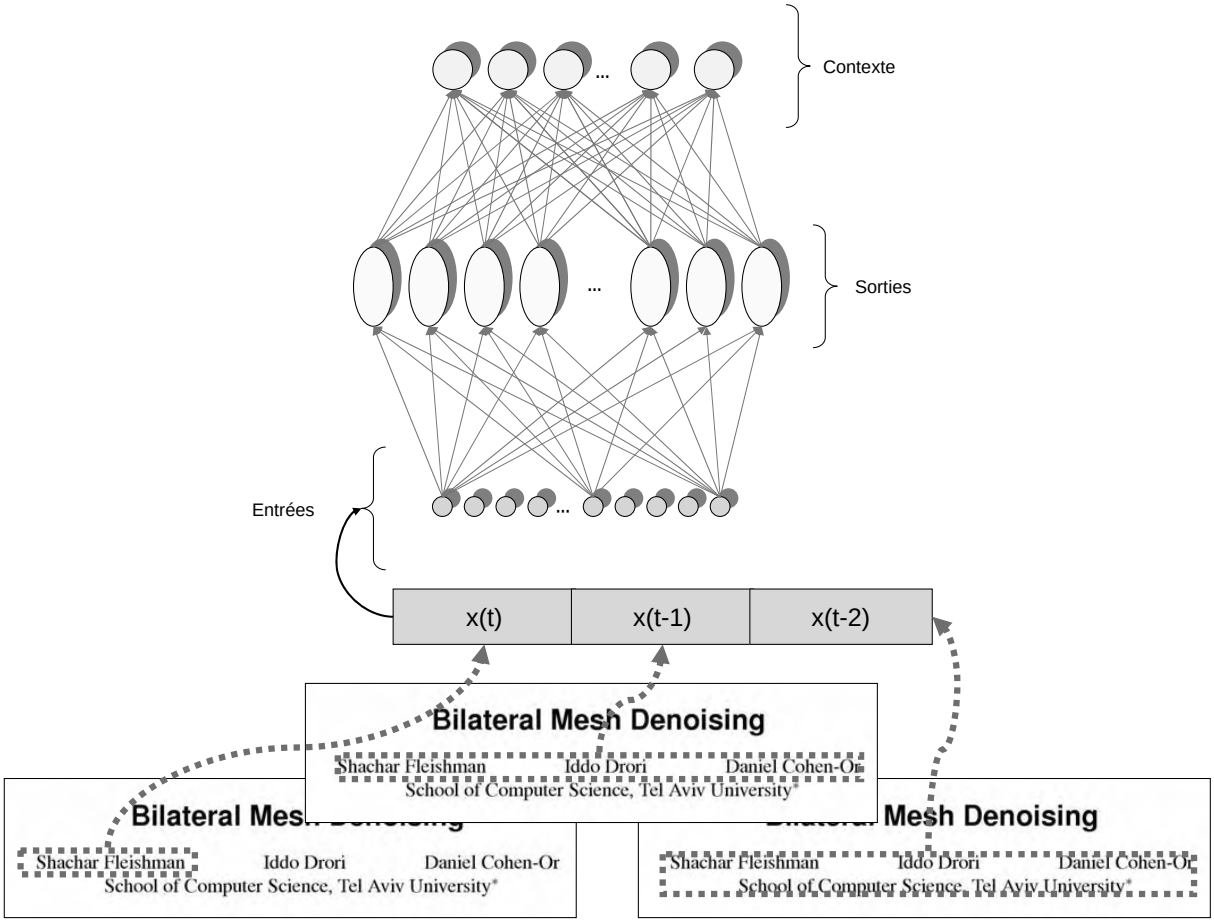


FIGURE 5.3 – Topologie du réseau de neurones dynamique perceptif

Les données recueillies sont donc entrées progressivement dans la ligne de temporisation. La première extraction donnera $x(t = 0)$, après un premier cycle perceptif, elle deviendra $x(t-1 = 0)$ et la nouvelle extraction pour le cycle perceptif suivant sera mise dans $x(t = 1)$ et ainsi de suite.

Le choix de la taille D de la ligne de temporisation est plus délicat : il faut estimer combien de «mémoire» doit être conservée. Cela revient à estimer le nombre maximum de cycles perceptifs à effectuer pour que les entrées de chaque cycle aient une place dans la ligne de temporisation. De nos expérimentations, il ressort que ce nombre est très petit : au bout du troisième cycle, sur la base des articles scientifiques, les résultats n'évoluent presque plus. D'autres cycles sont toujours possibles mais le rapport entre le gain de reconnaissance et le nombre de nouvelles extractions à refaire est trop minime. Dans les expérimentations de [Côté, 1997] et [Snoussi Maddouri, 2003], il fallait une dizaine de cycles mais comme leur réseau est très différent du nôtre, il ne travaille pas sur les mêmes données et comme leur fonction d'activation doit atteindre la saturation, on peut aussi estimer que, chez eux, le nombre de cycles est aussi petit. Au vu des critiques imputables au RDT, nous avons préféré conserver une petite mémoire ($D = 3$) car le nombre de cycles ne sera que très rarement plus élevé. Si l'on décide de poursuivre les corrections des entrées après $t > 2$, nous garderons le même réseau et la même ligne de temps à trois entrées en déplaçant les donnés avec $t \leftarrow t - 1$ (Fig. 5.4). Nous aurons donc toujours les trois plus récentes extractions en entrée du RNDP, en perdant à partir du quatrième cycle la première extraction et ainsi de suite si d'autres cycles sont nécessaires.

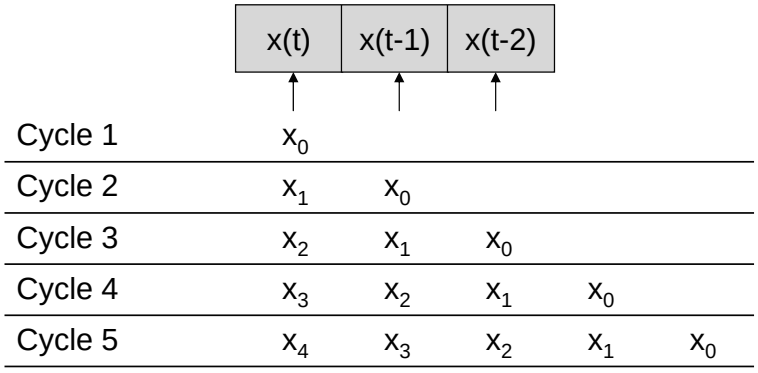


FIGURE 5.4 – Utilisation d'un réseau récurrent et intégration du retour de contexte dans le calcul de l'activation

Ce choix de D petit se justifie autrement que par l'aspect complexité calculatoire, dans le sens où, si plus de trois cycles perceptifs sont nécessaires à une forme, il est alors très probable que lors de ses premiers essais, le système se soit complètement trompé sur l'étiquette de la forme. Le retard pris dans les premiers cycles, sur la base d'une mauvaise hypothèse, sera finalement occulté par un D petit ; les extractions qui seront faites suite à un changement fructueux d'hypothèses seront toujours bien placées dans la ligne de temporisation. De plus, même dans la situation d'un simple affinement ou validation d'hypothèses, si le nombre de cycles requis dépasse la constante D, les conséquences devraient être minimes car la variation entre les entrées à deux instants consécutifs sera suffisamment modeste pour que le décalage $t \leftarrow t - 1$ n'ait pas de réelle influence sur la réponse du réseau.

L'apprentissage, vu en section 5.3.2 est utilisé sans changements. Il nécessite cependant d'avoir des entrées pour chaque instant, de 1 à D. La sortie attendue d(t) ne varie pas au cours du temps dans notre application, l'étiquette du bloc restant constante. La base d'apprentissage statique ne peut donc pas convenir à l'apprentissage du réseau pour $D > 1$. Il faut donc créer différentes extractions pour $t = \{1, 2, 3\}$ si l'on décide de fixer $D = 3$. Il suffit simplement de garder par exemple l'historique des changements effectués par l'opérateur lors de l'élaboration de chaque document de vérité. Comme l'expert corrige de manière plus efficace, produisant un document parfait en peu de retours sur données, nous pensons, pour cette raison supplémentaire, que D doit être petit, du moins se rapprochant plus du nombre moyen de cycles à effectuer pour reconnaître les formes plutôt que de choisir par exemple le nombre maximum. Un D petit donnera des résultats plus tranchés et incitera plus à la correction lors d'une ambiguïté plutôt qu'à une validation d'hypothèses sur plusieurs cycles.

Le partitionnement des données n'intervenant que sur la taille de l'entrée, il restera lui aussi valide. Il convient cependant de prendre en considération, lors de l'apprentissage, toutes les éventualités possibles. En effet, après une propagation, le système peut soit demander une correction des entrées actuelles, soit ajouter un nouveau groupe de variables. Au cycle suivant, le RNDP doit potentiellement travailler avec deux ensembles différents de variables. Pour éviter de devoir explorer toutes les combinaisons possibles, une solution serait de forcer le réseau à utiliser toujours dans le même ordre et au même moment les différents groupes de variables. Dans nos expérimentations, nous avons donc fixé à la fois le nombre de sous-ensembles à trois (Chap. 4) et, comme évoqué précédemment, la taille de la ligne de temporisation aussi à trois. Les trois premiers cycles se feront donc toujours en débutant avec trois sous-ensembles croissants de variables, le dernier étant l'ensemble complet.

L'introduction du temps dans le réseau de RNP est une amélioration non négligeable comme nous le verrons dans la section consacrée aux expérimentations (Sec. 5.5). Le fait d'apprendre la correction à l'apprentissage permet de faire les bons choix d'insertion d'hypothèses ou de modification de boîte englobante plus tôt ou plus efficacement. Le temps d'apprentissage est plus conséquent mais le temps de reconnaissance reste lui toujours dans des délais très raisonnables et largement très inférieur au temps nécessaire à l'extraction des variables. D'autres perfectionnements seraient envisageables : nous avons évoqué les réseaux récurrents en sous-section 5.2.1 qui pourraient être une alternative ou un ajout très bénéfique au RNDP comme nous allons le montrer dans la section suivante.

5.5 Expérimentations

Nous reprenons les mêmes bases et les mêmes protocoles qu'en section 3.3 et 4.5. Le réseau de neurones dynamique perceptif est utilisé pour ces expérimentations, la taille de la ligne de temporisation est fixée à $D = 3$ tout comme le nombre de cycles perceptifs. Le tableau 5.1 présente les résultats obtenus par le RNDP en comparaison avec une version statique et un PMC.

Dans des conditions similaires, la version dynamique gagne environ 2,8% de reconnaissance supplémentaire. Elle est même plus performante que la version statique à son quatrième cycle avec le même facteur de temps qu'à son troisième cycle. L'utilisation des résultats des anciens cycles est donc bénéfique dans le cadre de la reconnaissance de structures logiques, on note une amélioration de la performance du système sans accroître le temps d'exécution. L'apprentissage est par contre beaucoup plus lent ; les calculs sont plus nombreux, le fait de travailler sur des

	Perceptron multicouche	Réseau de neurones statique perceptif	Réseau de neurones dynamique perceptif
Taux rec.	81,7%	90,2%	92,7%
Facteur temps	1	1,85	1,8

TABLEAU 5.1 – Classification de structures logiques par un PMC, un réseau de neurones perceptif et son extension dynamique, avec cycles perceptifs et partitionnement de l'espace d'entrée

vecteurs ralentit encore plus l'ensemble du processus. En comparaison avec un PMC, le RNDP, nécessite dix fois plus de temps pour son apprentissage. Lors de la reconnaissance, la propagation est toujours quasiment instantanée, c'est une fois de plus l'extraction des observations physiques qui prédomine et représente entièrement le facteur temps. Le RNPD est d'ailleurs même sensiblement plus rapide au troisième cycle que la version statique car certaines formes sont reconnues plus tôt ce qui permet d'éviter certaines extractions inutiles.

D'un point de vue qualitatif, les formes non reconnues sont pratiquement les mêmes que celles qui étaient aussi rejetées par les précédentes versions du réseau. Cela confirme la nécessité d'avoir recours à des outils d'extraction plus fiables et plus adaptés à la classe de document visée. Il n'en reste pas moins que le gain de reconnaissance obtenu en comparaison avec le PMC est significatif et que l'ensemble de la méthode apporte une réelle valeur ajoutée.

Si l'on observe de plus près les résultats pour chaque classe (Tab. 5.2), on remarque premièrement que pour les classes déjà bien reconnues par la version dynamique, on obtient cette fois-ci plusieurs classes avec un taux de reconnaissance de 100%. Ce sont globalement les classes qui se détachent, par leur aspect, le plus des autres. Les remerciements et la conclusion ne sont pas mieux reconnus, ce qui laisse supposer une fois de plus que les indices physiques dont nous disposons ne sont pas assez informatifs pour séparer ces deux classes de la classe des paragraphes.

La couche de contexte obtient des taux de reconnaissance aux cycles 1, 2 et 3 de respectivement 96,7%, 98,5% et de 99,6%. Elle est donc très fiable et permet de lever un grand nombre d'ambiguïtés et spécifiquement au premier cycle perceptif. Environ deux tiers des corrections de segmentation s'effectuent durant le premier cycle et permet d'approcher très vite les résultats d'un PMC. Si l'on se réfère par exemple aux travaux de [Krishnamoorthy et coll., 1993], qui utilisent une base d'article semblable à la notre, mais en utilisant le même nombre de structures logiques (Titre, Auteur, Numéro de page, Résumé, Mots-clés, Copyright, Section, Algorithme et Flottant), nous obtenons un taux de reconnaissance de 96,3% qui est supérieur aux 94,4% obtenus dans leurs travaux.

Classes	Nb. échantillons	Taux PMC	Taux RNDP avec cycles
Titre Document	15	93,3%	100,0%
Auteur	44	88,6%	93,2%
Email	5	0,0%	100,0%
Adresse	21	47,6%	85,7%
Résumé	15	93,3%	100,0%
Mots-clés	14	92,8%	92,9%
Catégories	9	88,9%	100,0%
Introduction	73	80,8%	80,8%
Paragraphe	440	96,1%	97,3%
Section	92	97,8%	97,8%
Sous-Section	62	98,3%	98,4%
Sous-sous-section	17	76,4%	82,4%
Liste	69	97,1%	98,6%
Énumération	44	95,4%	97,7%
Flottant	105	91,4%	99,1%
Conclusion	38	28,9%	28,9%
Bibliographie	187	98,9%	100,0%
Algorithme	86	95,3%	98,8%
Copyright	9	88,8%	100,0%
Numéro page	30	96,6%	96,7%
Remerciements	10	70,0%	70,0%

TABLEAU 5.2 – Résultats détaillés du réseau de neurones dynamique perceptif pour chaque classe

5.6 Perspectives

Le réseau à décalage temporel permet de reconnaître des séries d'observations dans le temps et nous avons exploité sa forte similitude avec le Perceptron multicouche pour l'utiliser directement dans notre RNP. Le réseau étant capable de tenir compte d'entrées évoluant au cours du temps, nous pouvons désormais l'employer à reconnaître une forme se basant sur les observations actuelles mais aussi celles provenant de cycles perceptifs antérieurs. Le RNDP est donc plus adapté à reconnaître ses entrées corrigées.

Le réseau de neurones perceptif proposé a été élaboré pour se rapprocher au mieux des observations et des déductions faites par [McClelland et Rumelhart, 1981]. Nous avons donc décidé d'utiliser aussi une version *feedforward* bien qu'en toute logique, lors de la reconnaissance, nous fassions des retours de contexte et qu'une version *feedback* serait peut-être plus appropriée. Pour nous asseoir sur des bases fortes proposées par Côté et Snoussi Maddouri, nous avons modifié le réseau de telle sorte à apporter des améliorations tout en gardant les principes cognitifs défendus par ces auteurs. Une version *feedback*, comme illustrée par la figure 5.5, serait capable de se servir de l'information contenue dans les couches de contexte pour la faire intervenir au

niveau du calcul de l'activation.

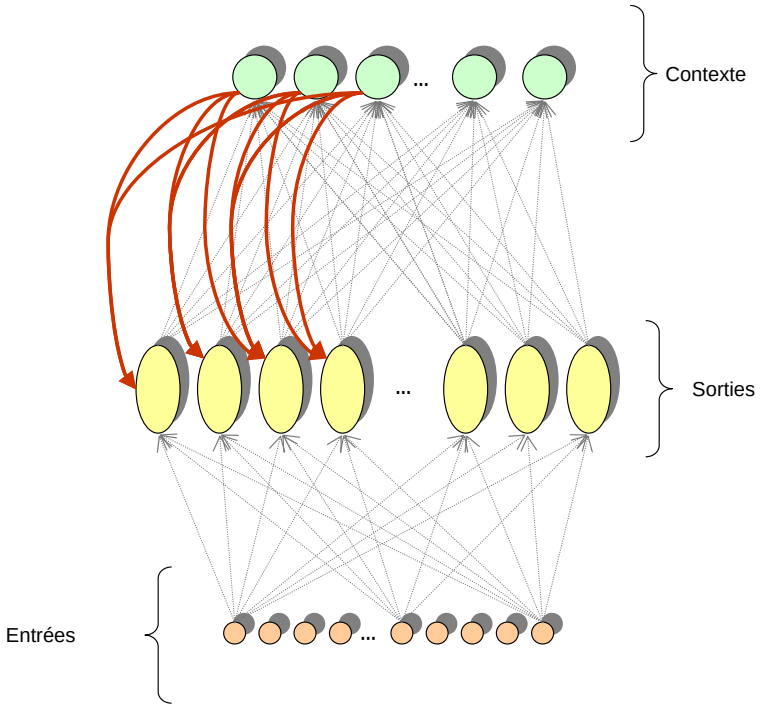


FIGURE 5.5 – Version récurrente avec retour d'état dans les couches précédentes

Avec les principes vus en section 5.2, il serait tout à fait envisageable d'avoir à la fois une architecture récurrente et temporelle. L'apprentissage serait certainement plus long et plus difficile à paramétrer, mais comme le réseau serait de taille raisonnable, qu'il serait toujours organisé en couches et que chaque neurone aurait une signification, nous nous retrouverions dans une situation assez particulière qui simplifierait l'ensemble du processus.

L'inconvénient de l'introduction de la récurrence dans le réseau serait de s'éloigner des principes psycho-cognitifs sur les fondements desquels nous avons élaboré le RNDP. Les liens ne seraient peut-être plus aussi faciles à interpréter, les sorties plus difficiles à analyser et la correction pourrait être biaisée à cause de la trop grande influence du contexte sur les couches inférieures. Donner plus de « poids » au réseau est aussi un risque de s'éloigner d'une version où le contexte et les décisions sont pris normalement en dehors du réseau. Nous pensons, comme [Nagy, 2000], que l'intégration de connaissances apparaît essentielle pour résoudre une tâche d'analyse de la structure logique. De part le cadre applicatif sur lequel nous appliquerons la méthode, il ne nous semble pas nécessaire d'utiliser une version récurrente pour améliorer les résultats de reconnaissance. Nous pensons que l'approche doit effectivement être orientée par les données comme nous le faisons avec l'architecture neuronale, mais se doit aussi de conserver une orientation par le modèle assez importante comme le propose la littérature. L'étude du passage à une version récurrente reste toutefois une perspective qui pourrait être très prometteuse et qui a d'autres propriétés intéressantes pour les données que nous manipulons.

Les réseaux de neurones récurrents ont en effet aussi d'autres applications qui nous confortent dans l'idée de faire intervenir une telle architecture dans notre système. Dans les travaux de [Küchler et Goller, 1996], les auteurs détournent l'utilisation d'un réseau récurrent, normalement prévu pour reconnaître des séquences temporelles, au cas de l'analyse de formes

structurées. Les formes structurées sont celles qui peuvent être représentées par un graphe acyclique orienté, étiqueté et à racine *(RLDAG, rooted labeled directed acyclic graph)*, où il existe un nœud n'ayant aucune entrée ou, par extension de la méthode, à n'importe quel graphe direct acyclique (Fig. 5.6).

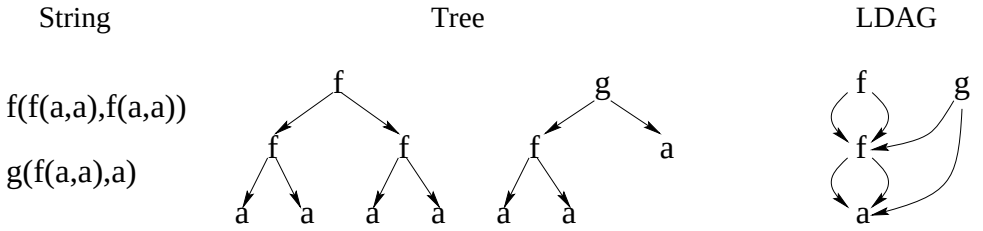


FIGURE 5.6 – Représentation d'un ensemble de termes par, de gauche à droite, une chaîne, un arbre et un graphe étiqueté direct acyclique

L'objectif de leurs travaux est de développer une architecture neuronale capable de retrouver la structure d'origine sachant un ensemble de RLDAG comme étant des instances de la structure. D'un point de vue statique, ils déplient le graphe sur un PMC ; les premières couches s'occupent de transformer le graphe en une entrée facilement interprétable par un PMC, les autres couches s'occupant de la tâche d'approximation ou de classification (Fig. 5.7).

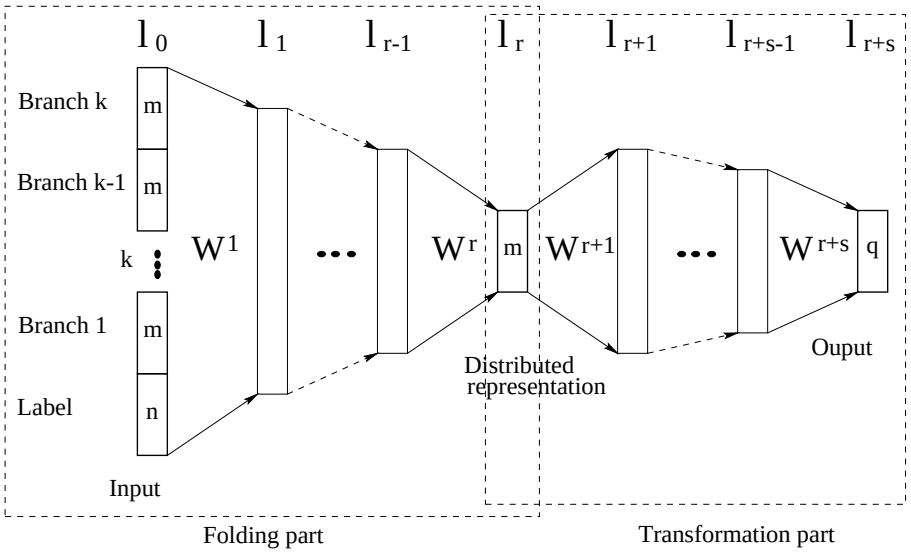


FIGURE 5.7 – Architecture générale du réseau de [Küchler et Goller, 1996]

La dynamique du réseau est définie par :

$$o_j^l(t) = f\left(\sum_i o_i^{l-1}(t) + w_{ij}^l + \theta_j^l\right) \tag{5.24}$$

où $o_i^l(t)$ est la sortie du neurone i dans la couche l à la récursion t, θ_i^l le biais associé au neurone i de la couche l, w_{ij}^l le poids de la connexion entre le neurone i de la couche l et le neurone j de la couche $l+1$ et j la fonction sigmoïde. L'apprentissage fait par la rétropropagation à travers la structure est une extension de la rétropropagation à travers le temps. L'erreur à minimiser est donnée par :

$$E = \sum_{i=1}^{p} \sum_{j=0}^{q-1} \frac{1}{2} \left([T_i]_j - o_j^{r+s}(root(s_i)) \right)^2 \tag{5.25}$$

où *root* est la fonction donnant le nœud père de s_i, et t_i la sortie attendue ($t_i = g(s_i)$, g étant la fonction à approcher). Les expérimentations ont été effectuées sur la classification de termes logiques à deux classes où ils obtiennent en moyenne 98%.

L'idée de travailler sur des données structurées peut aussi être une perspective intéressante pour notre réseau de neurones perceptif. Comme la structure logique est elle-même un arbre, il serait envisageable d'utiliser une partie du raisonnement de [Küchler et Goller, 1996] bien qu'il n'ait pas les mêmes objectifs car il n'étiquette pas son arbre mais classe juste les formes selon leurs structures.

Des travaux similaires ont été entrepris par [Sperduti et Starita, 1997] qui proposent une généralisation du travail des précédents auteurs avec le concept de «neurone complexe récursif». Le but est là aussi de trouver une fonction permettant de mettre en correspondance un domaine structuré avec un ensemble de réels. La sortie du neurone est définie par :

$$o(x) = f\left(\sum_{i=1}^{N_L} w_i l_i + \sum_{j=1}^{out\_degree_X(x)} \hat{w}_j o(out_X(x,j)) \right) \tag{5.26}$$

avec w_i les poids pondérant le vecteur d'entrée, N_L le nombre d'entrées encodant le label l tel que $l = g(x)$, g la fonction d'étiquetage et $\hat{w}$ les poids des connexions récursives. Tout comme dans les travaux de [Küchler et Goller, 1996], le réseau se décompose en une partie d'encodage (Fig. 5.8) et une autre de classification afin de pouvoir revenir à un réseau complètement *feedforward* (Fig. 5.9).

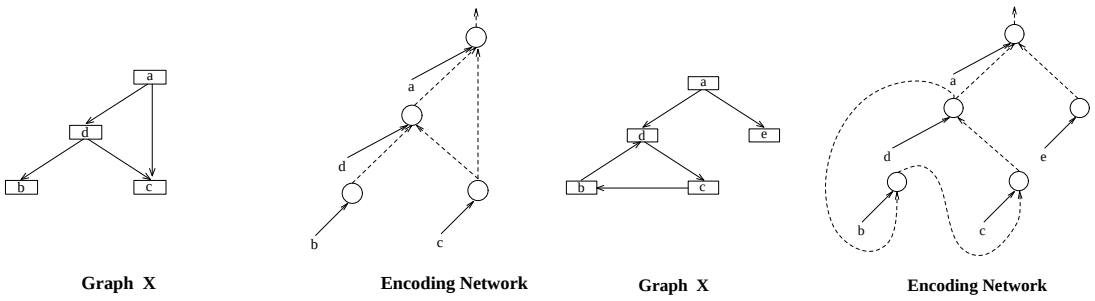


FIGURE 5.8 – Réseau d'encodage pour un graphe acyclique et pour un graphe avec cycle

L'algorithme d'apprentissage peut se faire par une extension de la rétropropagation à travers le temps mais aussi par l'apprentissage récurrent en temps réel [Williams et Zipser, 1989] et par d'autres algorithmes dynamiques comme les modèles LRAMM [Sperduti et coll., 1995], la

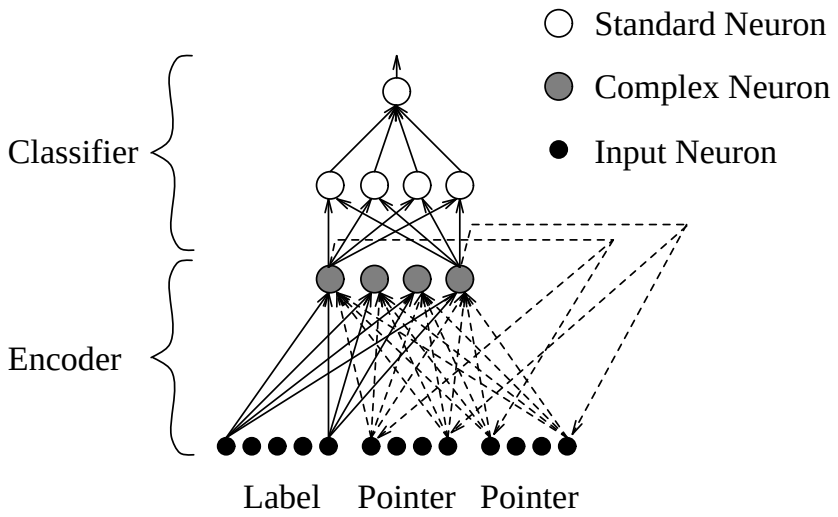


FIGURE 5.9 – Proposition d'architecture pour la classification des structures par [Sperduti et Starita, 1997] : les neurones complexes récursifs génèrent une représentation neuronale des structures qui sont ensuite classifiées par un réseau *feedforward*

corrélation en cascade pour les structures [Fahlman et Lebiere, 1990] et les réseaux de neurones arborescents [Sankar et Mammone, 1992].

Leur réseau est capable de traiter les graphes acycliques orientés tout comme ceux comportant des cycles. Les tests sont eux aussi semblables à ceux des précédents auteurs et ils se servent de plusieurs fonctions servent à générer des exemples positifs de graphes. Les résultats sont tout aussi bons : souvent 100% sur certains problèmes avec une moyenne d'environ 98%.

Bien qu'ici aussi, les tests soient effectués dans le domaine de la logique des termes et sur des données synthétiques, la méthode proposée reste applicable à de nombreux réseaux ou algorithmes d'apprentissage et peut travailler avec des formes structurées très générales (jusqu'au graphe avec cycle). Ces méthodes n'ont jamais été à notre connaissance appliquées à l'analyse de documents. Il est vrai qu'une fois de plus, le fait de reconnaître les formes structurées se prête davantage à des techniques à base de grammaires (la structure logique étant elle-même une production d'une grammaire) ; il est d'ailleurs assez difficile d'évaluer les résultats fournis car les données sont synthétiques. La robustesse de l'approche par le neurone complexe récursif est encore à vérifier dans des cas réels comme ceux que nous traitons. Quoi qu'il en soit, la possibilité de pouvoir une fois de plus déplier le réseau en une partie pour l'encodage et une partie pour la reconnaissance nous laisse supposer qu'il serait possible d'étendre, sans trop de changements, le réseau de neurones perceptif à une version structurée.

5.7 Conclusion

Nous avons décrit au cours de ce chapitre une amélioration du réseau de neurones perceptif (RNP) en proposant l'utilisation d'une architecture dynamique afin de prendre en compte, à l'intérieur même du réseau, la variabilité des entrées après chaque cycle perceptif. Après avoir

exposé un état de l'art des principaux réseaux dynamiques, nous avons justifié le choix de l'utilisation du réseau à décalage temporel (RDT) comme solution au problème de la dépendance des entrées au temps.

Le RDT étant basé sur un Perceptron multicouche, aucune modification profonde de la topologie n'a du être nécessaire pour conserver l'architecture à représentation semi-locale du RNP. L'extension à la version dynamique se fait à l'aide de la ligne de temporisation qui sert de mémoire au réseau. Initialement prévue pour reconnaître des séquences temporelles, nous avons employé cette faculté pour tenir compte de la séquence des différentes entrées générées par les cycles perceptifs.

L'apprentissage reste similaire à celui du PMC, la rétropropagation temporelle effectue une descente de gradient sur l'erreur totale quadratique du réseau en faisant l'hypothèse que la matrice des poids reste constante entre deux instants. Le nouveau réseau dynamique (RNDP) peut alors tenir compte, lors de la phase de reconnaissance, de l'entrée actuelle mais aussi de celles des précédents cycles. Nous avons fixé la taille de la série d'observations à trois qui correspond aussi au nombre de sous-ensembles que nous générons par la méthode de partitionnement. De part la petite taille de la ligne de temporisation et le faible nombre de neurones présents, le comportement du réseau (en complexité ou en convergence) reste très proche du RNP ; l'apprentissage et la reconnaissance sont plus lents mais pour cette dernière, le temps mis pour effectuer une propagation reste toujours négligeable devant le temps d'extraction des observations physiques. La contrainte technique liée à ce nouveau réseau vient du fait qu'il faille une base d'apprentissage elle aussi étiquetée à travers le temps pour permettre une meilleure reconnaissance.

L'introduction de la dimension temporelle est un atout certain pour la méthode, les tests effectués dans ce chapitre ont montré que, dans notre cas expérimental, à nombre de cycles perceptifs équivalent, la version dynamique gagne en efficacité. Le réseau est capable de fournir une réponse plus appropriée en fonction du numéro du cycle perceptif et des données déjà acquises jusqu'alors. Une extension du RNDP à une version récurrente pourrait aussi apporter un regain de performance supplémentaire : l'utilisation de connexions *feedback* entre les neurones du contexte vers la couche des sorties pourrait elle aussi contribuer à mieux intégrer directement dans le réseau les corrections apportées après chaque cycle. Bien que ce choix nous éloignerait encore plus du réseau de [McClelland et Rumelhart, 1981], cette voie semble tout de même exploitable et prometteuse. Au vu de la nature arborescente de la structure logique et des travaux réalisés par [Sperduti et Starita, 1997] pour la reconnaissance de formes structurées, l'intégration d'une récurrence à l'intérieur du réseau conforte l'intérêt que nous portons aux réseaux de neurones récurrents.

Conclusion et perspectives

Nous avons abordé au cours de cette thèse le problème de la reconnaissance de structures logiques d'images de documents. Le challenge réside dans le fossé, souvent sous-estimé, qu'il existe entre, d'une part, les informations physiques que l'on peut extraire de l'observation de l'image et, d'autre part, l'information logique qu'il est possible de rattacher à chaque élément du texte. D'emblée, le lien entre les structures physiques et logiques ne semble pas poser de difficulté particulière car, si des conventions typographiques sont rigoureusement suivies, l'apparence du document est un résultat connu à partir de la décomposition logique. Il n'est donc pas étonnant que la littérature abonde de méthodes à base de règles et de systèmes experts car il est naturel, pour l'analyse d'une image, d'utiliser une méthode similaire à celle qui a permis sa génération. Malheureusement, s'il est vrai que la conversion du logique vers le physique peut s'écrire en quelques règles, l'opération inverse est loin d'être triviale car il n'existe pas de bijection entre les deux. Plusieurs autres techniques ont été utilisées pour remédier à ce problème en utilisant la nature hiérarchique des deux structures ; les représentations en arbre et les systèmes à base de grammaires occupent eux aussi une place importante. Ces dernières sont plus flexibles et traitent des ensembles plus vastes de documents mais elles ne sont pas toujours performantes lorsque l'image est dégradée, rendant l'extraction de la structure physique erronée ou incomplète. Les méthodes dirigées par le modèle ne sont donc pas une solution générique au problème de l'extraction de structures logiques. A contrario, les approches par les données sont rarement utilisées pour cette tâche. On les retrouvera, en proportion inverse, dans les étapes précédant l'analyse logique dans lesquelles elles obtiennent de très bons résultats. Le faible engouement pour ce type de méthode tient en partie au fait que ces systèmes demandent une phase d'apprentissage fastidieuse et ne sont au final pas plus performants que ceux dirigés par le modèle. Il sera donc plus fréquent de rencontrer des systèmes à base de grammaires utilisant une phase d'apprentissage plutôt qu'une approche neuronale complètement dédiée à la tâche d'analyse.

L'introduction du modèle et de la connaissance semble donc être une quasi-nécessité pour mener à bien le problème. Il semblerait aussi qu'utiliser le raisonnement inverse, à savoir introduire le modèle dans une approche dirigée par les données, n'ait jamais été développé jusqu'alors. Nous avons fait le choix d'explorer cette voie tout en gardant à l'esprit que seule une hybridation entre les deux familles d'approches pourrait nous mener à une solution permettant de traiter les formes ambiguës que l'on trouve dans les documents, que ce soit à cause de la mauvaise qualité des informations physiques disponibles ou de la complexité inhérente au document. Bien que partant de problèmes différents, des constatations similaires ont été faites dans d'autres domaines comme celui de la reconnaissance de l'écriture cursive. En se basant sur les travaux de [McClelland et Rumelhart, 1981], [Côté, 1997] et [Snoussi Maddouri, 2003],

il nous est apparu que le réseau de neurones développé par ces auteurs possède les capacités nécessaires pour résoudre une partie du problème soulevé. Son fonctionnement à représentation locale, son principe de reconnaissance par cycles perceptifs et son utilisation du contexte permettant de revenir sur les données d'entrée qui provoquent une mauvaise reconnaissance sont autant de points facilitant la reconnaissance. Cette façon de procéder, basée sur une approche perceptive, nous a paru être un point d'ancrage intéressant et nous avons décidé de faire de ce réseau le point de départ de notre propre système appelé réseau de neurones dynamique perceptif.

En plus des adaptations nécessaires au niveau de la topologie, nous avons apporté une première contribution par l'apport d'un apprentissage qui était inexistant dans les précédentes versions. Par la même occasion, nous avons restauré les connexions inhibitrices proposées par les premiers auteurs et nous laissons l'apprentissage s'occuper seul de décider de la présence et de l'intensité des poids. La structure logique est dépliée à travers les couches du réseau et chaque neurone est porteur d'un concept. L'effet de la supériorité du mot sur la lettre a été remplacé par le contexte apporté par la hiérarchie de la structure logique : des concepts de plus en plus généraux et englobants sont placés sur les dernières couches du réseau. Les cycles perceptifs et la correction de la segmentation les accompagnant ont été conservés mais, à la différence des précédents auteurs, notre correction extrait de nouvelles observations physiques, au lieu de présenter dans un ordre différent les données, et les critères de rejet utilisés sont plus adaptés à la nouvelle fonction d'activation choisie.

Le changement de la correction donne lieu à une augmentation du taux de reconnaissance mais entraîne inévitablement un accroissement du temps de traitement. Pour réduire les extractions inutiles, nous avons proposé l'utilisation de groupes de variables à chaque cycle afin de ne présenter, dans un premier temps, qu'un sous-ensemble réduit de variables et de classifier une première partie des formes, puis d'utiliser les autres variables, progressivement, en fonction de la complexité de la forme. Le partitionnement des données est réalisé par une méthode par filtre, reprenant les premières phases de l'analyse en composantes principales, qui a été modifiée pour minimiser la redondance à l'intérieur de chaque ensemble. Les gains sont effectifs, même dans une situation où le temps d'extraction de chaque variable est sensiblement équivalent. Cette technique est d'autant plus intéressante si des traitements lourds mais peu fréquents sont nécessaires pour lever l'ambiguïté sur une forme.

Afin d'optimiser au mieux la réponse du réseau de neurones en fonction de l'avancement des cycles perceptifs, nous avons fait évoluer le réseau statique vers une version dynamique. L'intégration du fonctionnement d'un réseau à décalage temporel dans notre topologie permet de tenir compte de la variabilité des données d'entrée qui sont susceptibles d'être différentes après chaque cycle. En nous appropriant le concept de ligne de temporisation, nous pouvons prendre une décision en sachant les résultats antérieurs obtenus. Le nouveau réseau de neurones proposé adapte sa réponse en fonction du numéro du cycle perceptif et des changements qu'a subis la forme au cours des précédentes corrections. Bien que l'apprentissage du réseau dynamique soit beaucoup plus lent que celui de la version statique, la propagation n'est que sensiblement ralentie et est toujours bien inférieure au temps mis par l'extraction des observations physiques. La reconnaissance s'en trouve améliorée et le temps total de reconnaissance est par la même occasion sensiblement réduit. Le passage à la version dynamique est donc bénéfique en temps et en performance et demande uniquement un surcroît de traitement durant l'apprentissage qui ne se fait qu'une seule fois.

Le réseau de neurones dynamique perceptif a été testé sur une base d'articles scientifiques comprenant un nombre de structures logiques assez conséquent. Comparé à une approche pu-

rement neuronale, il obtient des scores largement supérieurs avec une fonction d'acceptation plus stricte que celle utilisée généralement et dans des temps raisonnables. Les scores obtenus sont comparables à ceux des méthodes dirigées par les données bien que le nombre de classes que nous utilisons soit largement supérieur à ce que propose la littérature. Nous partons aussi de données réelles, provenant principalement d'un seul OCR généraliste. Il est donc évident qu'en utilisant à la fois des observations plus pertinentes et des outils d'extraction plus efficaces, les résultats sont facilement améliorables. Tout comme dans les autres méthodes vues au cours du manuscrit, les résultats absolus dépendent de la qualité de l'extraction physique. Les gains relatifs, obtenus par rapport à une approche neuronale classique, sont eux aussi perfectibles. Plusieurs directions théoriques ont été évoquées au cours des trois derniers chapitres. Le partitionnement des données a été fait de manière totalement automatique, d'autres méthodes plus coûteuses et demandant plus de paramétrages empiriques sont envisageables. Au vu de l'état de l'art conséquent dans ce domaine, nous opterions pour une méthode à base d'algorithme génétique. La création de groupes n'est pas en soi capitale dans le sens où, pour certaines formes, il sera nécessaire d'utiliser toutes les variables disponibles. Il n'empêche que la manière dont sont constitués les groupes peut avoir une conséquence sur le temps de reconnaissance, voire sur la qualité des résultats à nombre de cycles fixé.

La perspective la plus intéressante est celle concernant l'architecture du réseau. Comme évoqué au cours du dernier chapitre, l'utilisation d'une version récurrente serait bénéfique lors de l'utilisation des cycles perceptifs pour profiter au mieux du contexte. De façon générale, la topologie du réseau peut être modifiée de différentes manières afin d'être plus en adéquation avec le problème traité. Il serait possible par exemple d'introduire simplement des couches cachées entre chaque niveau d'interprétation pour obtenir une meilleure flexibilité du système. Bien que nous ayons voulu conserver au maximum les fondements théoriques des systèmes de nos prédécesseurs, il serait intéressant de perdre éventuellement l'interprétabilité du réseau et une facilité lors des retours de contexte pour espérer obtenir une amélioration en ayant un réseau à représentation distribuée. Pour en revenir aux réseaux récurrents, les travaux de [Küchler et Goller, 1996] et [Sperduti et Starita, 1997 ; Frasconi et coll., 1998] sont des voies intéressantes à suivre car la structure logique est elle aussi représentable par un graphe étiqueté. Nous avons aussi vu qu'il était possible de manipuler à la fois des réseaux dynamiques et récurrents. Il est donc très probable que l'intégration d'une récurrence dans le RNDP ne soit pas un obstacle tout comme le passage de la version statique à la version dynamique et que cet axe de recherche nous semble être le plus prometteur si l'on se concentre uniquement sur le classifieur.

Annexes

Annexe A

La base des articles scientifiques

La base de documents utilisée est constituée d'articles scientifiques provenant de la conférence Siggraph[11]. Les échantillons présentés ici (de A.1 à A.6) sont des extractions directes des PDF fournis par la conférence. Les images réellement utilisées sont élaborées en imprimant premièrement les fichiers avec une imprimante jet-d'encre, puis en numérisant à l'aide d'un copieur le lot de documents papier. Les images de documents sont à une résolution de 600 DPI en noir et blanc. Aucun prétraitement n'est effectué sur les images. La figure A.7 donne un aperçu de la différence entre la qualité du document vectoriel PDF et l'image obtenue après numérisation.

[11] ACM SIGGRAPH, Special Interest Group on GRAPHics and Interactive Techniques, `http://www.siggraph.org/s2003/`

Interactive Boolean Operations on Surfel-Bounded Solids

Bart Adams* Philip Dutré*

Department of Computer Science
Katholieke Universiteit Leuven

Abstract

In this paper we present an algorithm to perform interactive boolean operations on free-form solids bounded by surfels. We introduce a fast inside-outside test to check whether surfels lie within the bounds of another surfel-bounded solid. This enables us to add, subtract and intersect complex solids at interactive rates. Our algorithm is fast both in displaying and constructing the new geometry resulting from the boolean operation.

We present a resampling operator to solve problems resulting from sharp edges in the resulting solid. The operator resamples the surfels intersecting with the surface of the other solid. This enables us to represent the sharp edges with great detail.

We believe our algorithm to be an ideal tool for interactive editing of free-form solids.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations; I.3.6 [Computer Graphics]: Methodology and Techniques—Graphics data structures and data types I.3.4 [Computer Graphics]: Graphics Utilities—Graphics editors

Keywords: free-form modeling, boolean operations, surfels, point-based geometry

1 Introduction

Constructive solid geometry (CSG) has been a useful tool in computer graphics for many years. Usually, CSG is applied to primitive objects (spheres, cylinders, cubes) to construct objects with a more complex geometric shape. However, boolean operations are also a versatile tool for editing free-form solids. Adding, subtracting and intersecting solids enables us to create more complex models. In this paper we present boolean operations as an intuitive and interactive editing tool for free-form solids bounded by surfels. Surfels, represented as oriented points in 3D space, approximate the local orientation of the surface they represent. Each surfel can be considered to represent a small area of this surface. As a consequence, when performing boolean operations on two solids A and B, most of the surfels of the surface of solid A are completely inside or outside solid B and vice versa. Only a small number of surfels intersect with the surface of the other solid.

Our algorithm works in two steps: in a first step we classify the surfels of both solids as inside, outside or intersecting with the surface of the other solid. In a second step we resample the surfels intersecting with the surface of the other solid. Our method is fast, both in displaying the boolean operations as in calculating the new geometry of the resulting solid. An example of a free-form surfel-bounded solid constructed with our algorithm is shown in figure 1.

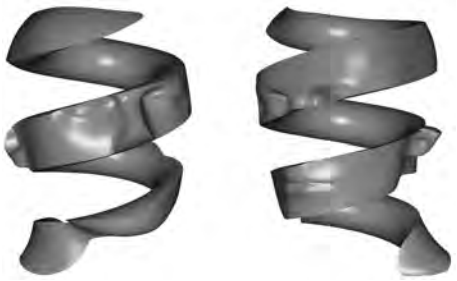


Figure 1: Two free-form surfel-bounded solids constructed using CSG (inspired by "Bond of Union" by M.C. Escher).

This paper addresses the following important questions:

- How to test efficiently whether a surfel of one surfel-bounded solid lies inside or outside another surfel-bounded solid?
- How to represent the sharp edges typically resulting from performing boolean operations on solids, using surfels?

We start by giving a brief overview of related work in section 2. Section 3 introduces the concepts related to surfel-bounded solids. In section 4 we present a fast inside-outside test that enables us to classify the surfels of solid A as inside, outside or intersecting with the surface of solid B. In section 5 we consider the surfels intersecting with the surface of the other solid and propose the fast resampling operator. Section 6 gives implementation details and illustrates that we are able to perform boolean operations on complex solids at interactive rates. We conclude and give some topics of future research possibilities in section 7.

2 Related Work

Point-Based Geometry

The interest in using points as a display primitive in computer graphics has grown tremendously in recent years. Pfister et al. [2000] introduced the concept of surfels inspired by the work of Levoy and Whitted [1985], and more recently the work of Grossman and Dally [1998]. Significant research has been performed on efficient high quality rendering of point-based geometry. QSplat [Rusinkiewicz and Levoy 2000] uses a hierarchy of bounding spheres for progressive rendering of large models. Zwicker et al. [2001] introduce surface splatting which makes the benefits of the Elliptical Weighted Average (EWA) filter available to point-based rendering. Alexa et al. [2001] present point set surfaces and use down-sampling and up-sampling to meet the required display quality. Kalaiah and Varshney [2001]

*email:{barta,phil}@cs.kuleuven.ac.be

FIGURE A.1 – Page 1 du premier article

Operation	Surface of A kept	Surface of B kept
$A \cup B$	outside B	outside A
$A \cap B$	inside B	inside A
$A - B$	outside B	inside A
$B - A$	inside B	outside A

Table 1: Part of surfaces kept when performing boolean operations.

use differential points that capture the local differential geometry in the vicinity of the sampled point. More recently Botsch et al. [2002] introduce a compact representation that uses less than two bits per point position. Cohen et al. [2001] and Chen and Nguyen [2001] introduced a hybrid system, combining polygon and point rendering. Recent approaches [Ren et al. 2002; Coconu and Hege 2002] exploit the power of current graphics hardware to render point-based geometry with high quality.

Also, work has been published on modeling and editing point-sampled geometry. Pauly and Gross [2001] introduce spectral filtering and resampling of point-based geometry. Pointshop 3D [Zwicker et al. 2002] extends 2D photo editing to 3D point clouds. They introduce a set of tools (painting, sculpting and filtering) to edit the geometry and appearance of the model. However, geometry modeling is limited to normal displacement. Pauly et al. [2002] are able to perform large model deformations on point-based geometry thanks to a dynamic resampling strategy.

Constructive Solid Geometry

Lots of research has been performed concerning constructive solid geometry. For an excellent overview we refer to [Foley et al. 1996] and [Hoffmann 1989]. Interactive rendering of CSG is often performed using graphics hardware [Goldfeather et al. 1986; Goldfeather et al. 1989; Rappoport and Spitz 1997]. Another method for CSG display is to convert the CSG structure to a boundary representation which can be rendered by all rendering systems. Interactive modification of boundary representations is often slow and difficult. Recent work however has proven that it is possible to compute the result of boolean operations on free-form solids in a reasonable amount of time. Kristjansson et al. [2001] present a framework to perform boolean operations on free-form solids bounded by multiresolution subdivision surfaces. Museth et al. [2002] present a level set framework to perform various surface editing operations.

We extend their work to solids bounded by surfels. We present boolean operations on surfel-bounded solids as an interactive editing tool. The work presented in this paper is mostly related to the work of Kristjansson et al. and Museth et al. Our algorithm can not only display the result of the boolean operation, but also compute the resulting solid at interactive rates. We also show that we are able to represent the sharp features in the resulting solid.

3 Surfel-Bounded Solids

The objects used in this paper are closed solids whose surface is represented by surfels. Each surfel s consists of a position $\mathbf{x}_s$, a radius of influence r_s and an orientation $\mathbf{n}_s$. Therefore surfels can be thought of as disks orthogonal to $\mathbf{n}_s$ with center $\mathbf{x}_s$ and radius r_s. The radius r_s should be chosen so that the projections of the disks on the image plane overlap. The surfel-bounded solids are obtained by LDC (*layered depth cube*) sampling and 3-to-1 reduction as described in [Pfister et al. 2000]. Initially each surfel will thus have a radius $r_s = \sqrt{3}h$ with h the sampling distance in each dimension chosen to match the required display resolution. Although we use uniformly sampled solids, our algorithms do not rely on this. For each solid we define $r_{max} = \max r_s$ as the radius of the largest surfel belonging to its surface.

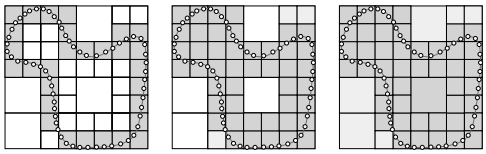


Figure 2: Constructing the quadtree of depth $d = 3$. Left: in a first step the quadtree is constructed; the blue cells are the boundary cells. Middle: classifying the empty leaf cells at depth $d = 3$. Green cells are inside the solid, yellow cells are outside the solid. Right: classifying the empty leaf cells at depth $d - 1$, i.e. 2.

4 Inside-Outside Test

When constructing a new surfel-bounded solid from two solids A and B we have to determine which surfels of A and B will be part of the surface of the resulting solid. Depending on the boolean operation different parts of the surfaces of A and B will represent the boundary of the new solid. E.g. when taking the difference $A - B$ we want to keep the part of the surface of A that is outside B and the part of the inverted surface of B that is inside A. Table 1 gives an overview for the different boolean operations.

In this section we propose a fast inside-outside test that enables us to classify the surfels of solid A as inside, outside or intersecting with the surface of solid B and vice versa. The inside-outside test is based on 3-color octrees [Samet 1990] with leaf cells classified as interior, exterior or boundary. For boundary leaf cells we partition the space even further using two parallel planes.

For clarity the ideas presented in this section are illustrated in two dimensions, but are easily extended to 3D.

4.1 Octree Construction

For each solid we construct an axis-aligned octree. We start with the bounding box containing all the surfels of the solid and subdivide it into 8 equally sized children. Each node is recursively split into 8 children as long as it contains surfels and as long as a user-chosen depth d (typically 4 or 5) is not reached.

After constructing the octree, the empty cells are classified as being inside or outside the solid, as illustrated in figure 2. The resulting octree has three types of leaf cells: boundary cells, empty cells inside the solid and empty cells outside the solid. Within a node of the octree, each cell has a neighbor in one of the principal directions. If an empty cell has a non-empty neighbor we look at the orientation of the surfels in this neighboring cell. The orientation of the surfel that is closest to the empty cell tells us if the empty cell is inside or outside the solid: if this surfel is pointing towards the empty cell, the empty cell must be outside the solid, if the surfel is pointing away from the empty cell, the empty cell must be inside the solid. More formally: let $\mathbf{c}_e$ and $\mathbf{c}_n$ be the coordinates of the centers of the empty cell and its non-empty neighbor and let s be the surfel closest to the empty cell with normal $\mathbf{n}_s$, then the empty cell is classified as inside if $(\mathbf{c}_n - \mathbf{c}_e) \cdot \mathbf{n}_s > 0$. Otherwise, the empty cell must be outside the solid.

There are three different cases when classifying an empty cell:

- the empty cell has only one non-empty neighbor,
- the empty cell has more than one non-empty neighbor (figure 3, left),
- the empty cell has no non-empty neighbor (figure 3, right).

In the first case the empty cell is classified by looking at this non-empty neighbor. In the second case, we only consider one of the non-empty neighbors. In the third case, we first classify the neighbors, and give the same classification to the empty cell as neighboring empty cells must have the same classification. Because a node in the octree has at least one non-empty cell, we can always

FIGURE A.2 – Page 2 du premier article

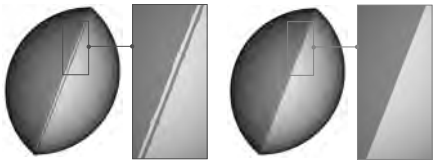


Figure 7: Intersection of two spheres. Left: no resampling. Right: resampling of surfels which intersect with the other surface. This results in sharp edges without significant overshoot, even under magnification.

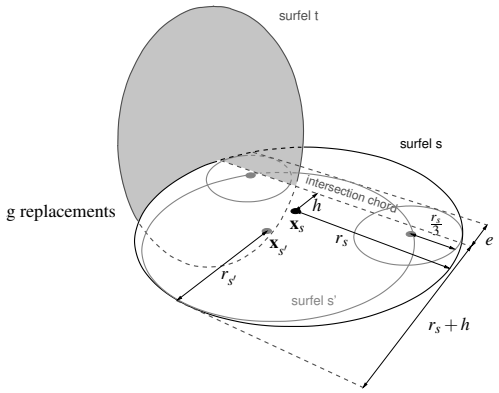


Figure 8: Intersection of the surfel s and the plane through its nearest neighbor t in the other solid. Surfel s is replaced by three smaller surfels by the resampling operator.

References

ADAMS, B., AND DUTRÉ, P. 2003. A smoothing operator for boolean operations on surfel-bounded solids. Tech. rep., April.

ALEXA, M., BEHR, J., COHEN-OR, D., FLEISHMAN, S., LEVIN, D., AND SILVA, C. T. 2001. Point set surfaces. *IEEE Visualization 2001* (October), 21–28.

BOTSCH, M., WIRATANAYA, A., AND KOBBELT, L. 2002. Efficient high quality rendering of point sampled geometry. In *Proceedings of the 13th workshop on Rendering*, Eurographics Association, 53–64.

CHEN, B., AND NGUYEN, M. X. 2001. Pop: a hybrid point and polygon rendering system for large data. In *IEEE Visualization 2001*, 45–52.

COCONU, L., AND HEGE, H.-C. 2002. Hardware-accelerated point-based rendering of complex scenes. In *Proceedings of the 13th workshop on Rendering*, Eurographics Association, 43–52.

COHEN, J. D., ALIAGA, D. G., AND ZHANG, W. 2001. Hybrid simplification: combining multi-resolution polygon and point rendering. In *IEEE Visualization 2001*, 37–44.

FOLEY, J. D., VAN DAM, A., FEINER, S. K., AND HUGHES, J. F. 1996. *Computer graphics (2nd ed. in C): principles and practice*. Addison-Wesley Longman Publishing Co., Inc.

GOLDFEATHER, J., HULTQUIST, J. P. M., AND FUCHS, H. 1986. Fast constructive-solid geometry display in the pixel-powers graphics system. In *Computer Graphics (Proceedings of SIGGRAPH 86)*, vol. 20, 107–116.

GOLDFEATHER, J., MOLNAR, S., TURK, G., AND FUCHS, H. 1989. Near real-time csg rendering using tree normalization and geometric pruning. *IEEE Computer Graphics & Applications 9*, 3 (May), 20–28.

GOTTSCHALK, S. 1996. Seperating axis theorem. Tech. Rep. TR96-024, Dept. of Computer Science, UNC Chapel Hill.

GREENSPAN, M., GODIN, G., AND TALBOT, J. 2000. Acceleration of binning nearest neighbor methods. In *Proceedings of Vision Interface 2000*, 337–344.

Head		Helix			
number of surfels	octree depth	number of surfels	octree depth	interaction time	update time
30k	4	60k	4	130 ms (7.7 FPS)	900 ms
90k	5	170k	5	240 ms (4.2 FPS)	2150 ms
200k	5	250k	5	340 ms (2.9 FPS)	2890 ms
350k	5	370k	5	500 ms (2 FPS)	4690 ms

Table 2: Timings for the head-helix difference for different numbers of surfels and octree depths.

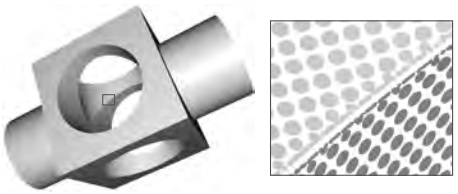


Figure 9: The classic CSG example. The cube consists of 65k surfels, the cylinder of 50k surfels. Average interaction rate is 16 FPS. Average update time is 600 ms. Right: closeup drawn using smaller disks (radii $r_s/2$) for the surfels.

GROSSMAN, J. P., AND DALLY, W. J. 1998. Point sample rendering. In *Eurographics Rendering Workshop 1998*, 181–192.

HOFFMANN, C. M. 1989. *Geometric and solid modeling: an introduction*. Morgan Kaufmann Publishers Inc.

KALAIAH, A., AND VARSHNEY, A. 2001. Differential point rendering. In *Rendering Techniques 2001: 12th Eurographics Workshop on Rendering*, 139–150.

KRISTJANSSON, D., BIERMANN, H., AND ZORIN, D. 2001. Approximate boolean operations on free-form solids. In *Proceedings of ACM SIGGRAPH 2001*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, 185–194. ISBN 1-58113-292-1.

LEVOY, M., AND WHITTED, T. 1985. The use of points as a display primitive. Tech. Rep. TR85-022, January.

MUSETH, K., BREEN, D. E., WHITAKER, R. T., AND BARR, A. H. 2002. Level set surface editing operators. *ACM Transactions on Graphics 21*, 3 (July), 330–338.

PAULY, M., AND GROSS, M. 2001. Spectral processing of point-sampled geometry. In *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, 379–386.

PAULY, M., KOBBELT, L., AND GROSS, M. 2002. Multiresolution modeling of point-sampled geometry. Tech. rep., September.

PFISTER, H., ZWICKER, M., VAN BAAR, J., AND GROSS, M. 2000. Surfels: Surface elements as rendering primitives. In *Proceedings of ACM SIGGRAPH 2000*, ACM Press / ACM SIGGRAPH / Addison Wesley Longman, Computer Graphics Proceedings, Annual Conference Series, 335–342. ISBN 1-58113-208-5.

RAPPOPORT, A., AND SPITZ, S. 1997. Interactive boolean operations for conceptual design of 3-d solids. In *Proceedings of SIGGRAPH 97*, Computer Graphics Proceedings, Annual Conference Series, 269–278.

REN, L., PFISTER, H., AND ZWICKER, M. 2002. Object space ewa surface splatting: A hardware accelerated approach to high quality point rendering. *Computer Graphics Forum 21*, 3, 461–470. ISSN 1067-7055.

RUSINKIEWICZ, S., AND LEVOY, M. 2000. Qsplat: A multiresolution point rendering system for large meshes. In *Proceedings of ACM SIGGRAPH 2000*, ACM Press / ACM SIGGRAPH / Addison Wesley Longman, Computer Graphics Proceedings, Annual Conference Series, 343–352. ISBN 1-58113-208-5.

SAMET, H. 1990. *The design and analysis of spatial data structures*. Addison-Wesley Longman Publishing Co., Inc.

ZWICKER, M., PFISTER, H., VAN BAAR, J., AND GROSS, M. 2001. Surface splatting. In *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, 371–378.

ZWICKER, M., PAULY, M., KNOLL, O., AND GROSS, M. 2002. Pointshop 3d: An interactive system for point-based surface editing. *ACM Transactions on Graphics 21*, 3 (July), 322–329. ISSN 0730-0301 (Proceedings of ACM SIGGRAPH 2002).

FIGURE A.3 – Page 5 du premier article

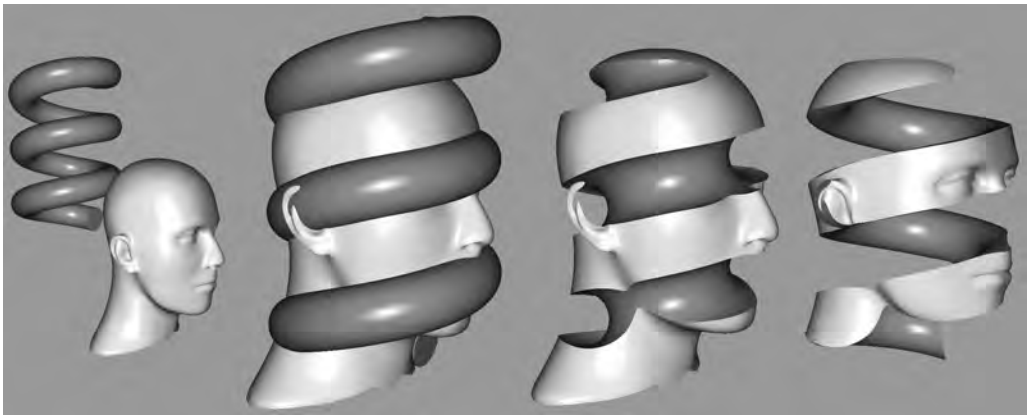


Figure 10: Constructing one head for the Bond of Union (after M.C. Escher) from the mannequin head (350k surfels, octree depth 5) and a helix (370k surfels, octree depth 5). Resulting geometry of union, difference and intersection are shown. During interactive manipulation we obtain a frame rate of 2 frames per second.

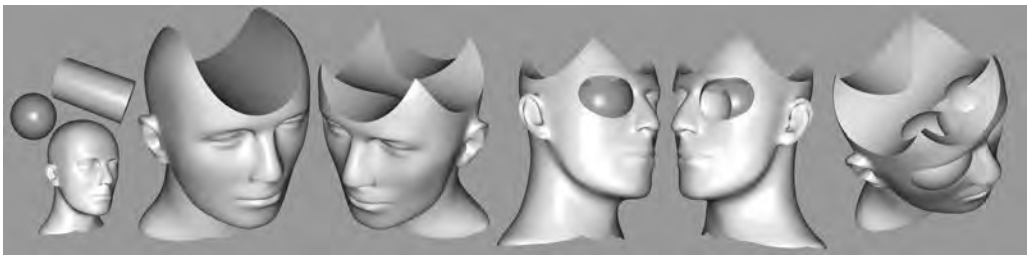


Figure 11: Subtracting 2 cylinders (230k surfels each, octree depth 4) and 2 spheres (46k surfels each, octree depth 4) from the mannequin head (350k surfels, octree depth 5). Average interaction rate is 4.4 FPS. Left: original solids. Middle: the four boolean operations. Right: resulting geometry.

Figure 12: Mythical centaur constructed from the horse model (340k surfels), the Venus model (250k surfels) and the dragon model (650k surfels) all with octree depth 5. Average interaction time of union between horse model and Venus model was 2.5 FPS, between dragon head and centaur body was 3.3 FPS. Local smoothing is performed in the neighborhood of the surface-surface intersections.

FIGURE A.4 – Page 6 du premier article

Structured Importance Sampling of Environment Maps

Sameer Agarwal* Ravi Ramamoorthi† Serge Belongie* Henrik Wann Jensen*

*University of California, San Diego

†Columbia University

Importance w/ 300 samples | Importance w/ 3000 samples | Structured importance w/ 300 samples | Structured importance w/ 4.7 rays/pixel

Figure 1: *Close-up rendering of a glossy buddha in the grace cathedral environment. The two images on the left have been rendered using stratified importance sampling with 300 and 3000 samples, while the two images on the right show the result of structured importance sampling using 300 samples, and after further rendering optimizations an average of 4.7 rays per pixel to evaluate the 300 possible samples.*

Abstract

We introduce *structured importance sampling*, a new technique for efficiently rendering scenes illuminated by distant natural illumination given in an environment map. Our method handles occlusion, high-frequency lighting, and is significantly faster than alternative methods based on Monte Carlo sampling. We achieve this speedup as a result of several ideas. First, we present a new metric for stratifying and sampling an environment map taking into account both the illumination intensity as well as the expected variance due to occlusion within the scene. We then present a novel hierarchical stratification algorithm that uses our metric to automatically stratify the environment map into regular strata. This approach enables a number of rendering optimizations, such as pre-integrating the illumination within each stratum to eliminate noise at the cost of adding bias, and sorting the strata to reduce the number of sample rays. We have rendered several scenes illuminated by natural lighting, and our results indicate that structured importance sampling is better than the best previous Monte Carlo techniques, requiring one to two orders of magnitude fewer samples for the same image quality.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Shading

Keywords: Rendering, Image Synthesis, Illumination, Ray Tracing, Monte Carlo Techniques, Shadow Algorithms, Global Illumination, Environment Mapping

1 Introduction

To capture realistic natural lighting, it is common to use environment maps, a representation of the distant illumination at a point. Environment map rendering has a long history in graphics, going back to seminal work by Blinn and Newell [1976], Miller and Hoffman [1984], Greene [1986] and Cabral et al. [1987], as well as recent work on high-dynamic range imagery by Debevec [1998], and extensions of the basic environment mapping ideas by Cabral et al. [1999], Kautz and McCool [2000], Kautz et al. [2000], Ramamoorthi and Hanrahan [2001; 2002], and others.

Most of the previous environment mapping techniques [Miller and Hoffman 1984; Greene 1986; Ramamoorthi and Hanrahan 2001] are intended for real-time applications, and ignore visibility. They usually require an expensive pre-computation or pre-filtering step, where an irradiance environment map is obtained by convolving the incident illumination with the Lambertian or more complex reflection function. Ramamoorthi and Hanrahan [2001; 2002] propose fast pre-filtering methods using spherical harmonics, but their methods also make the common assumption of no cast shadows. In recent work, Sloan et al. [2002] have demonstrated real-time rendering taking visibility effects into account, but their technique is limited to static scenes with low-frequency lighting, and requires a slow pre-computation step involving ray tracing and detailed sampling of visibility.

In this paper, we address the problem of efficiently rendering high quality images of scenes illuminated by arbitrary environment maps. Our method specifically optimizes the integration of distant illumination on surfaces with Lambertian and semi-glossy BRDFs, it correctly accounts for occlusion within the scene (such as shadows due to bright lights in the environment map), and it handles scenes with changing geometry. In terms of global illumination research, our method can be viewed as an efficient technique for sampling millions of distant lights corresponding to pixels in an environment map. We seek to estimate the integral of a product of the visibility and the illumination. One of these, the illumination, is known, and is the same for every surface point in the scene, and may also be reused for multiple scenes or multiple frames of an animation. Therefore, unlike many previous image synthesis problems, it is feasible to perform extensive preprocessing on the environment map without degrading performance. Visibility, on the other hand can be complicated and changes throughout the scene, requiring sampling for general scenes. Naive Monte Carlo sampling such as

path tracing [Kajiya 1986] is well-suited for sampling visibility, but in the presence of high-frequency environment maps it results in significant noise, since it does not take the variation of the illumination into account. In this situation it is better to use importance sampling based on the illumination in the environment map. Even though importance sampling is significantly better than path tracing, it is not deterministic and results in significant sampling noise as shown in Figure 1. Pure illumination based importance sampling also tends to use too many samples on small bright lights such as the sun in the blue sky even though it is very small, within which the variation in visibility is mostly insignificant.

To understand how to sample an environment map, we present a novel analysis of visibility variance, and develop a metric for sampling both visibility and illumination efficiently. We also introduce a general and automatic hierarchical stratification algorithm for partitioning environment maps into a set of area light sources. The algorithm performs hierarchical thresholding of the map and uses our importance metric to deterministically allocate samples to each level. The samples are then placed inside each level using the Hochbaum-Shmoys clustering algorithm [Hochbaum and Shmoys 1985], which has strong runtime and quality guarantees associated with it.

Our stratification algorithm ensures a good sampling pattern of the environment map, and in addition it enables a number of rendering optimizations that are difficult to include in standard Monte Carlo techniques. We can eliminate sampling noise and make the method completely deterministic by pre-integrating the illumination in each stratum — effectively collapsing the stratum into a directional light source. This results in a set of lights approximating the environment similar to the output of the LightGen plug-in for HDRShop [Cohen and Debevec 2001].

2 Monte Carlo Sampling and Importance

In this section, we analyze Monte Carlo integration of irradiance due to environment maps in more detail with the purpose of defining an appropriate importance metric. The irradiance, E, at a given surface location, x is computed as

$$E(x) = \int_{\Omega_{2\pi}} L_i(\vec{\omega}) S(x,\vec{\omega}) (\vec{\omega}\cdot\vec{n})\, d\vec{\omega}\,, \qquad (1)$$

where $\Omega_{2\pi}$ is the hemisphere of directions above x, L_i is the incident radiance or environment map, indexed only as a function of angle $\vec{\omega}$, S is the (binary) visibility in the direction $\vec{\omega}$, and $\vec{n}$ is the surface normal at x. For our analysis we will ignore the surface orientation and focus on the illumination and visibility. A complete rendering algorithm is presented in section 4. Our goal is to compute this integral efficiently using Monte Carlo sampling, where L_i is a known function that is easy to evaluate and the same for all points x, while S requires sampling since it is unknown and depends on x.

To understand how to distribute samples in the domain of the integrand, i.e. the illumination sphere, we introduce a new importance metric Γ. The two key competing strategies here are area-based stratified sampling and illumination-based importance sampling. To unify these two extremes and intermediate possibilities within a common framework, we use the following general metric for distributing samples in a region of solid angle $\Delta\omega$, with net (integrated) illumination L,

$$\Gamma(L,\Delta\omega) = L^a \Delta\omega^b\,. \qquad (2)$$

Here, a and b are parameters we seek to determine. First, consider the extreme cases: $a = 1, b = 0$ corresponds to standard illumination-based importance regardless of area (this technique over-samples small bright lights), and $a = 0, b = 1$ corresponds to

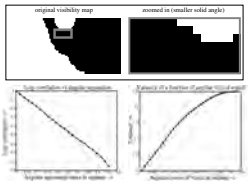


Figure 2: *Analysis of a representative visibility map, confirming the qualitative results expected, and validating our quantitative analysis.* **Top:** *Part of a visibility map for one pixel on the ground plane with a teapot casting shadows. On the left is the original binary visibility map. This region has approximately equal visible and shadowed regions and the variance over the whole region is the maximum, $1/4 = 0.25$. We zoom in on the red rectangle in the right figure. Even though this is one of the most complex regions in the original visibility map, it is clear that over a smaller solid angle, visibility is much simpler, and the variance drops down to less than 0.14.* **Bottom:** *Quantitative analysis of above effect. On the left, we plot the log of the correlation function $\alpha(\theta)$ as it varies with with the angular separation θ. Confirming our theoretical analysis, we find a straight line, showing that correlation decays exponentially with θ. The correlation angle T is estimated from this plot as $T \approx 0.5$. On the right, we plot the variance as a function of the angular separation. For small θ, the variance increases linearly, fairly accurately obeying Equation 6. As θ/T approaches $3/4$, the graph tails off, with the maximum variance of $1/4$ being approached.*

pure area-based stratification without considering illumination (this technique under-samples the bright lights). It would appear at first glance that both extremes have problems, and an intermediate parameter setting is better. To determine the parameters for the optimal metric we first analyze the variance due to visibility.

2.1 Variance Analysis for Visibility

We now present a novel preliminary theoretical and empirical analysis showing that variance in visibility is proportional to the angular extent (square root of solid angle) of the region in question, providing a basis for a new importance metric.

We analyze the expected variance of the visibility function $S(x,\vec{\omega})$ in a region subtending solid angle $\Delta\omega$, corresponding to a cluster or light source. Intuitively, we expect some coherence in the visibility function S, at least over small regions, so we expect the variance to be a function of $\Delta\omega$, with less variance in smaller regions and more in larger regions. Figure 2 shows some empirical tests on a representative visibility map—we have carried out experimental tests on approximately 10 visibility maps. In particular, in the top left, we show a relatively complex section of the visibility map (here, part of one face of a binary cubemap). Even in the most complex regions, the visibility function is much more coherent when zooming in on a smaller region, as shown in the top right of Figure 2.

For further quantitative analysis, we assume a correlation model for visibility. Mathematically, we can define a correlation function,

$$\alpha(\theta) = P(S(\vec{\omega}_1) = S(\vec{\omega}_2) \mid\mid \vec{\omega}_1 - \vec{\omega}_2 \mid\mid = \theta)\,, \qquad (3)$$

which simply measures the probability that the visibility function S is the same for two points separated by a distance (angle) θ. Assuming the worst case that the mean visibility $\bar{\beta}$ is $1/2$, or that overall $P(S=0) = P(S=1) = 1/2$, gives the highest overall vari-

Miller, G., and Hoffman, C. 1984. Illumination and reflection maps: Simulated objects in simulated and real environments. *SIGGRAPH 84 Advanced Computer Graphics Animation seminar notes.*

Nemhauser, G. L., and Wolsey, L. A. 1988. *Integer and Combinatorial Optimization*. John Wiley and Sons, Inc.

Ramamoorthi, R., and Hanrahan, P. 2001. An efficient representation for irradiance environment maps. In *SIGGRAPH 01*, 497–500.

Ramamoorthi, R., and Hanrahan, P. 2002. Frequency space environment map rendering. In *SIGGRAPH 02*, 517–526.

Sloan, P., Kautz, J., and Snyder, J. 2002. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *SIGGRAPH 02*, 527–536.

Veach, E., and Guibas, L. J. 1995. Optimally combining sampling techniques for monte carlo rendering. In *SIGGRAPH '95*, 419–428.

Ward, G. J., Rubinstein, F. M., and Clear, R. D. 1988. A ray tracing solution for diffuse interreflection. In *Computer Graphics (SIGGRAPH '88 Proceedings)*, J. Dill, Ed., vol. 22, 85–92.

Ward, G. 1991. Adpative shadow testing for ray tracing. In *Second Eurographics Workshop on Rendering*.

Appendix

Standard Monte Carlo theory tells us that the variance of a function (random variable) is given by

$$V[S] = E[S^2] - E[S]^2. \qquad (11)$$

Since shadows are binary, $S^2 = S$, and $E[S^2] = E[S]$. Letting $\bar{\beta} = E[S]$, we get $V[S] = \bar{\beta} - \bar{\beta}^2 = \bar{\beta}(1-\bar{\beta})$.

We want to compute the expected variance (for one sample) of the visibility function S for a small solid angle $\Delta\omega$. The expected value of the variance (where for clarity, we denote expected values over all regions by an overline) will be given by

$$\overline{V[S,\Delta\omega]} = \overline{E[S,\Delta\omega] - E[S,\Delta\omega]^2} = \overline{\bar{\beta}(\Delta\omega)(1-\bar{\beta}(\Delta\omega))}, \qquad (12)$$

where we define $\bar{\beta}(\Delta\omega)$ as the average or expected value of S over solid angle $\Delta\omega$.

The key expression above is $\bar{\beta}(\Delta\omega)(1-\bar{\beta}(\Delta\omega))$. Consider the case when $\bar{\beta} = 1$ (the entire region of interest is visible). The above expression is then 0, and there is no variance. Similarly, if $\bar{\beta} = 0$, the variance is 0. In fact, the worst case is $\bar{\beta} = 1/2$, corresponding to a variance of $1/4$. Over the entire image, we will assume the worst case of random visibility, $P(S=0) = P(S=1) = 1/2$.

We want to consider the expected variance as a function of solid angle $\Delta\omega$. The intuition is that as $\Delta\omega$ becomes smaller than the feature size of visibility, the probability distribution for $\bar{\beta}(\Delta\omega)$ is bimodal, i.e. either $\bar{\beta} = 0$ or $\bar{\beta} = 1$, and the corresponding variance tends to zero. Note that we must consider the average or expected value of the variance $\overline{\bar{\beta}(1-\bar{\beta})}$ and not separately $\bar{\beta} \times \overline{1-\bar{\beta}}$, which tends to $1/4$ as $\Delta\omega$ tends to 0, since $\bar{\beta}$ has equal probability of being 0 or 1.

To analyze further, we assume a correlation model for visibility, as per Equation 5. Now assume that the central point of the region of interest, subtending solid angle $\Delta\omega$, is visible, i.e. $S(0) = 1$. Analysis with $S(0) = 0$ is symmetric. We will assume that the visibility at a point making an angle θ with the central point is described using the simple correlation model above, so that the expected value, given $S(0) = 1$ is simply $\alpha(\theta) = \frac{1}{2}(1 + \hat{\alpha}(\theta))$. In other words,

$$E[S(\theta) \mid S(0) = 1] = \frac{1}{2}\left(1 + e^{-\theta/T}\right), \qquad (13)$$

which, as expected, tends to 1 as $\theta \to 0$, and tends to $1/2$ as θ becomes large.

Now, all that remains is to compute the expected variance. For this, it suffices to compute the expected value of $\bar{\beta}(\Delta\omega)$. The variance, under the assumption $S(0) = 1$ will simply be $\bar{\beta}(\Delta\omega)(1-\bar{\beta}(\Delta\omega))$. A similar argument holds for $S(0) = 0$, so this is the quantity we seek. Now, $\bar{\beta}(\Delta\omega)$ is the expected value of S over the whole solid angle $\Delta\omega$, which is the same as integrating the expected value of $S(\theta)$ at each point. Finally, the solid angle $\Delta\omega = \pi\theta^2$, assuming the maximum angle θ with respect to the central point is small.

$$\begin{aligned}\bar{\beta}(\Delta\omega) &= \frac{1}{\pi\theta^2}\int_0^\theta\int_0^{2\pi}\frac{1}{2}\left(1+e^{-u/T}\right)u\,du\,d\phi \\ &= \frac{1}{2} - \frac{T}{\theta}e^{-\theta/T} + \frac{T^2}{\theta^2}(1-e^{-\theta/T}). \end{aligned} \qquad (14)$$

Assuming, $\theta \ll T$ and taking the Taylor series expansion for $e^{-\theta/T}$ we obtain

$$\begin{aligned}\bar{\beta} &\approx \frac{1}{2} - \frac{T}{\theta}\left(1 - \frac{\theta}{T} + \frac{\theta^2}{2T^2}\right) + \frac{T^2}{\theta^2}\left(\frac{\theta}{T} - \frac{\theta^2}{2T^2} + \frac{\theta^3}{6T^3}\right) \\ &\approx 1 - \frac{\theta}{3T}. \end{aligned} \qquad (15)$$

Now, the variance is given by $\bar{\beta}(\Delta\omega)(1-\bar{\beta}(\Delta\omega))$ and this becomes

$$\overline{V[S,\Delta\omega]} \approx \frac{\theta}{3T}\left(1 - \frac{\theta}{3T}\right) \approx \frac{\theta}{3T}. \qquad (16)$$

Note that in these equations, the solid angle for small θ is given by $\Delta\omega = \pi\theta^2$, so $\theta = \sqrt{\Delta\omega/\pi}$, and the variance is proportional to the square root of the solid angle.

Finally, we seek to determine the optimal distribution of samples, and we will do so by attempting to minimize the variance of the net integral. For this, consider two regions of variance V_1 and V_2, with N samples to be distributed between them in the ratio ρN and $(1-\rho)N$. Assuming variance decreases at the rate of N^{-1},

$$V[N] = \frac{1}{N}\left(\frac{V_1}{\rho} + \frac{V_2}{(1-\rho)}\right), \qquad (17)$$

which we can differentiate with respect to ρ, obtaining

$$\frac{\rho}{1-\rho} = \sqrt{\frac{V_1}{V_2}}, \qquad (18)$$

and with more than two regions, this ratio must generally be followed, i.e. the number of samples is proportional to $\sqrt{V}$. As an example, consider $V = L^2$, assuming simple scaling of variance by illumination magnitude. In that case, the number of samples $\rho \sim L$, as for standard importance-based stratification. Now, consider the variance in a region of solid angle $\Delta\omega$. The visibility variance is proportional to $\Delta\omega^{1/2}$, but the net variance must be scaled by net illumination intensity L^2, assuming uniform lighting (this assumption also builds on our hierarchical thresholding scheme which creates levels of approximately equal illumination intensity). Hence, the number of samples is proportional to $\sqrt{L^2\Delta\omega^{1/2}} = L\Delta\omega^{1/4}$.

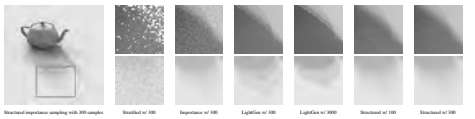


Figure 5: *A teapot in Galileo's tomb rendered using different sampling strategies. No sorting or jittering has been used for this comparison. The large image in the top row has been computed using structured importance sampling with 300 samples, which we verified to be indistinguishable from a reference image computed with 100,000 samples using standard Monte Carlo sampling. The red squares show two regions that have been rendered using different sampling techniques as close-ups in the small images on the right. From left to right these images have been rendered using naive stratified sampling, illumination based stratified importance sampling and using LightGen with 300 samples, LightGen with 3000 samples, structured importance sampling with 100 samples and 300 samples. Both Monte Carlo techniques produce significant statistical noise even for this simple model, LightGen shows banding in the shadows with both 300 and 3000 samples (since too few samples are placed at the bright lights), structured importance sampling looks convincing with just 100 samples and with 300 samples the result is indistinguishable from a reference image. The bottom row shows from left to right, the Galileo map, the lights created by LightGen, and the stratum centers created using our method. Note how our stratification method samples the bright lights much more densely than LightGen. This is the reason why the shadows with structured importance sampling are more accurate.*

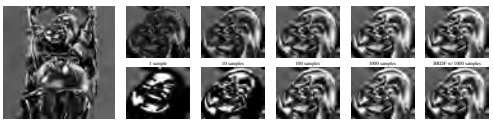


Figure 6: *A glossy buddha in the Grace environment map. The large image on the left is our sampling technique with 300 samples, which is practically indistinguishable from a reference image. The two rows show close-ups of the head rendered with an increasing number of samples. The top row is stratified importance sampling with 1, 10, 100 and 1000 samples as well as BRDF based importance sampling with 1000 samples. The bottom row shows structured importance sampling with 1, 10, 100, and 300 samples per pixel, as well as a version rendered with sorting and thresholding resulting in an average of just 4.7 samples per pixel. Note how structured importance sampling results in noise free images and quickly converges to the final result while the best Monte Carlo sampling techniques are noisy even when using 1000 samples.*

Figure 7: *Jittering can be used to eliminate banding at low sample counts at the cost of adding noise along the shadow boundaries. This image is the same close-up of the shadow as in Figure 5 using just 50 samples. The image on the left is without jittering and the image on the right has been rendered using jittering of the shadow ray.*

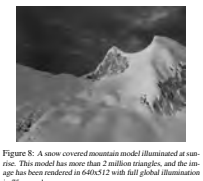
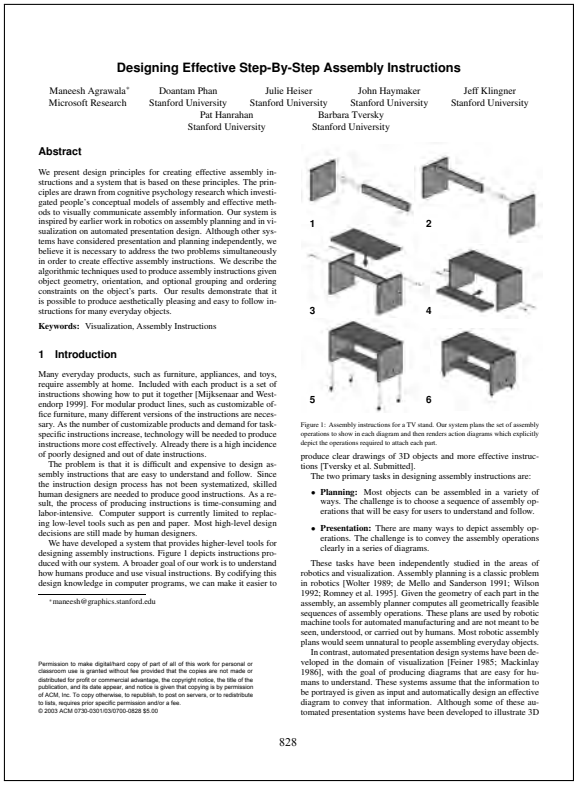


Figure 8: *A snow covered mountain model illuminated at sunrise. This model has more than 2 million triangles, and the image has been rendered in 640x512 with full global illumination in 75 seconds.*

FIGURE A.5 – Les deux premières et les deux dernières pages du second article scientifique



Designing Effective Step-By-Step Assembly Instructions

Maneesh Agrawala* Microsoft Research
Doantam Phan Stanford University
Julie Heiser Stanford University
John Haymaker Stanford University
Jeff Klingner Stanford University
Pat Hanrahan Stanford University
Barbara Tversky Stanford University

Abstract

We present design principles for creating effective assembly instructions and a system that is based on these principles. The principles are drawn from cognitive psychology research which investigated people's conceptual models of assembly and effective methods to visually communicate assembly information. Our system is inspired by earlier work in robotics on assembly planning and in visualization on automated presentation design. Although other systems have considered presentation and planning independently, we believe it is necessary to address the two problems simultaneously in order to create effective assembly instructions. We describe the algorithmic techniques used to produce assembly instructions given object geometry, orientation, and optional grouping and ordering constraints on the object's parts. Our results demonstrate that it is possible to produce aesthetically pleasing and easy to follow instructions for many everyday objects.

Keywords: Visualization, Assembly Instructions

1 Introduction

Many everyday products, such as furniture, appliances, and toys, require assembly at home. Included with each product is a set of instructions showing how to put it together [Mijksenaar and Westendorp 1999]. For modular product lines, such as customizable office furniture, many different versions of the instructions are necessary. As the number of customizable products and demand for task-specific instructions increase, technology will be needed to produce instructions more cost effectively. Already there is a high incidence of poorly designed and out of date instructions.

The problem is that it is difficult and expensive to design assembly instructions that are easy to understand and follow. Since the instruction design process has not been systematized, skilled human designers are needed to produce good instructions. As a result, the process of producing instructions is time-consuming and labor-intensive. Computer support is currently limited to replacing low-level tools such as pen and paper. Most high-level design decisions are still made by human designers.

We have developed a system that provides higher-level tools for designing assembly instructions. Figure 1 depicts instructions produced with our system. A broader goal of our work is to understand how humans produce and use visual instructions. By codifying this design knowledge in computer programs, we can make it easier to

*maneesh@graphics.stanford.edu

Figure 1: Assembly instructions for a TV stand. Our system plans the set of assembly operations to show in each diagram and then renders action diagrams which explicitly depict the operations required to attach each part.

produce clear drawings of 3D objects and more effective instructions [Tversky et al. Submitted].

The two primary tasks in designing assembly instructions are:

- **Planning:** Most objects can be assembled in a variety of ways. The challenge is to choose a sequence of assembly operations that will be easy for users to understand and follow.
- **Presentation:** There are many ways to depict assembly operations. The challenge is to convey the assembly operations clearly in a series of diagrams.

These tasks have been independently studied in the areas of robotics and visualization. Assembly planning is a classic problem in robotics [Wolter 1989; de Mello and Sanderson 1991; Wilson 1992; Romney et al. 1995]. Given the geometry of each part in the assembly, an assembly planner computes all geometrically feasible sequences of assembly operations. These plans are used by robotic machine tools for automated manufacturing and are not meant to be seen, understood, or carried out by humans. Most robotic assembly plans would seem unnatural to people assembling everyday objects.

In contrast, automated presentation design systems have been developed in the domain of visualization [Feiner 1985; Mackinlay 1986], with the goal of producing diagrams that are easy for humans to understand. These systems assume that the information to be portrayed is given as input and automatically design an effective diagram to convey that information. Although some of these automated presentation systems have been developed to illustrate 3D objects and actions [Seligmann and Feiner 1991; Rist et al. 1994; Butz 1997; Strothotte 1998], their primary focus has been on showing the locations or physical properties of parts.

Our approach is inspired by a combination of ideas from these previous systems. However, we believe that decisions involved in planning and presentation are strongly intertwined. Therefore both issues must be considered simultaneously.

The contributions of our work include:

Cognitive design principles for effective assembly instructions: We performed cognitive psychology experiments to identify how people conceive of the assembly process and to characterize the properties of well-designed instructions. Based on the results of these experiments and prior cognitive psychology research, we identify design principles for effective assembly instructions. These principles connect people's conceptual model of the assembly task to the visual representation of that task.

A system instantiating these design principles: Our assembly instruction design system consists of two parts: a planner and a presenter. The planner searches the space of feasible assembly sequences to find one that best matches the cognitive design principles. To do this the planner must also consider many aspects of presentation. The presenter then renders a diagram for each step of the assembly sequence generated by the planner. The presenter also uses the design principles to determine where to place parts, guidelines and arrows. In particular, the presenter can generate action diagrams which use the conventions of exploded views to clearly depict the parts and operation required in each assembly step.

2 Design Principles for Assembly Instructions

Before we can develop automated tools for designing assembly instructions, we must understand how people think about and communicate the process of assembling an object. Cognitive psychologists have developed a variety of techniques to investigate how people mentally represent ideas and concepts. We recently performed human subject experiments based on these techniques to determine the mental representations underlying assembly [Heiser and Tversky 2002]. We briefly describe our experimental setup.

In the first experiment, we asked participants to assemble a TV stand, given only a photograph of the completed stand as a guide. After they assembled the TV stand, we asked them to create a set of instructions that would show another person how to assemble it. Examples of the diagrams they drew are shown in Figure 2. In the second experiment, we asked a new group of participants to rank the effectiveness of a subset of the instructions produced in the first experiment. Finally, the third experiment tested whether the highly ranked instructions were more effective. Yet another group of participants used instructions ranked in the second experiment to assemble the TV stand, while experimenters recorded task completion time and error rates. We found that in general the highly ranked instructions were easier to understand and follow. Participants spent less time assembling the TV stand and made fewer errors.

Based on these experiments, as well as earlier cognitive research, we identify a set of design principles for creating assembly instructions that are easy to understand and follow.

Hierarchy and grouping of parts: People think of assemblies as a hierarchy of parts. At the base level, parts are segmented by perceptual salience indexed by contour discontinuity; that is, parts that are disjoint are more likely to be segmented. Typically, the disjoint parts are also grouped by different functions (e.g. the legs of a chair or the drawers of a desk) [Tversky and Hemenway 1984]. When possible, people prefer that parts within a group are added to the assembly at the same time, or in sequence one after another. The part groups are usually considered as hierarchical structures, which parallel the subassembly structure of the object.

Figure 2: Hand-drawn assembly diagrams for the TV stand. The action diagram is preferable to the structural diagram because it depicts the operations required to attach each part. In this case the action diagram shows how the shelf is fastened by the screws.

Hierarchy of operations: People think of the attachment operations required to build an assembly as a hierarchy of actions on the parts [Zacks et al. 2001]. At the higher levels, people consider the operations required to combine separate subassemblies. Our experiments showed that as people work down the subassembly hierarchy, they eventually consider the operations required to join significant individual parts. At the lowest level of the hierarchy, people consider attaching smaller parts and fasteners to the more significant parts. The significance of a part depends on a number of factors including function, size, and symmetry.

While the hierarchy of operations may contain many levels for complicated objects with numerous subassemblies (e.g. a car engine), we have found that a two-level hierarchy (significant parts and less important parts + fasteners) is common for many build-at-home objects, including most furniture. In this paper we focus on design tools for these two levels.

Step-by-step instructions: Our experiments confirmed the results of Novick et al. [2000] showing that people prefer instructions that present the assembly operations across a sequence of diagrams rather than a single diagram showing all the operations. Moreover, if the assembly contains significant parts as well as less important parts, people generally prefer that each diagram show how to attach only one significant part at a time. However, each diagram will usually show multiple non-significant part attachments. In Figure 1, the non-significant parts include the fasteners and the wheels.

While it is essential that the assembly diagrams are clear and easy to read, each diagram should also present as much information as possible. If instructions are split across too many diagrams, they become tedious to use. Similarly, some assemblies require the same sequence of operations to be repeated many times. For example, when assembling a bookcase, each shelf is attached in exactly the same way. Depicting such repetitive operations in detail can make the instructions unnecessarily long and tiresome. A better approach is to skip repetitive operations after they have been presented in detail a few times.

Structural diagrams and action diagrams: Based on analysis of the hand-drawn instructions we collected in the first experiment, we define two types of assembly diagrams: structural diagrams and action diagrams (see Figure 2). Structural diagrams present all the parts of the assembly in their final assembled positions; users must compare two consecutive diagrams to infer which parts are to be attached. Action diagrams spatially separate the parts to be attached from the parts that are already attached and use guidelines to indicate where the new parts attach to the earlier parts.

We found that action diagrams are superior to structural diagrams for the TV stand assembly task. We believe that this is because action diagrams contain all the information in the structural diagrams and also explicitly depict the attachment operations required in each step. However, toys such as LEGO often use structural diagrams rather than action diagrams. Showing the attachment operations may be less important because most LEGO parts fasten in the same way.

Orientation: Most objects have a set of natural orientations or preferred views [Palmer et al. 1981; Blanz et al. 1999]. These orien-

Figure 10: Assembly instructions for case27. The placement of the guidelines is automatically chosen to show how the parts attach to one another. The guidelines connect the centers of the bounding boxes of contact faces rather than connecting the centers of parts. In step 3 the guidelines properly show how the rectangular parts slide into the main hull.

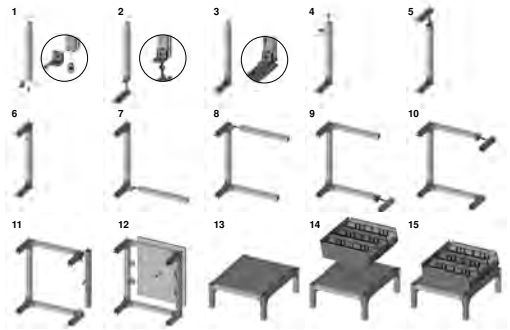


Figure 11: Assembly instructions for a table built from the 80/20 standardized parts. The first six steps detail how two of the L-shaped brackets are used to attach parts of the frame. The planner then omits these details in subsequent steps. Because the fasteners are much smaller than the other parts, they can be difficult to see in the original diagrams. We manually added the insets for the first 3 steps to make the fasteners easier to see.

only the first two bracket attachments (steps 1 through 6) are shown in full detail. In step 7 and beyond, the instructions no longer show the nuts and bolts required to secure the brackets to the frame. We manually added the insets for the first 3 steps to make the fasteners easier to see. Seligmann and Feiner [1991] have proposed an automated approach for designing such insets, and we are currently exploring the possibility of adding this technique to our system.

The performance of our system is presented in Table 1. The running time of the system is dominated by the low level geometric and visibility computations. We have not focused on optimizing these parts of the system and believe that more sophisticated low-level algorithms could increase the speed of that code. The final column of the table reports the number of subsets for which we compute the visibility score as well as the total number of subsets an exhaustive search would have to score. As Table 1 shows, our search optimizations significantly reduce the number of subsets scored.

7 Discussion

While our system can generate assembly instructions for a variety of objects, it also makes several basic assumptions that we hope to relax in future work.

Two-level hierarchy: The system operates on the two bottom levels of the hierarchy of operations (joining significant parts and attaching fasteners). Extending the system to handle subassemblies would allow for larger, more complicated assemblies.

Single-step translations along principal axes: We only consider single-step translational motions along the principal axes when computing the blocking relationships between parts. Using Guibas et al.'s [1995] approach to handle multi-step translations and rotations in all motion directions would increase the types of assemblies our system could handle.

Figure 12: Exploded view of case27. Our stack-building procedure can be used on the entire assembly rather than individual steps to produce such exploded views. The algorithm properly handles building stacks for different separation directions. As shown in Figure 10, the four rods at the top of the assembly slide into the paddles and rectangular parts below. After expanding the stacks, the rods no longer align with the parts they slide into. Therefore, the system does not generate guidelines showing how the rods attach to the assembly.

Local interference: Blocking relationships are computed for local pairs of parts that are in contact with one another. However, it is possible that parts which are not in contact block one another. Therefore, global interference detection would impose stronger, more robust feasibility constraints on the assembly sequence. Wilson [1992] has proposed techniques for computing this type of global interference.

Input of semantic/functional knowledge: Our system is designed to use semantic and functional knowledge about the parts when it is provided. In practice we have supplied this information manually. However, it may be possible to infer some of these properties from the part geometry based on models of perception. For example, it may be possible to automatically group parts that are perceived as roughly symmetric.

Although these assumptions do limit the types of assemblies our system can handle, we believe that the overall framework of the system is sound. Relaxing any of these assumptions would require localized changes to modules within the framework rather changes to the framework itself.

8 Conclusions

We have described a set of design principles for designing effective assembly instructions that are easy to understand and follow. The principles are based on cognitive psychology research examining how people mentally represent and communicate the process of assembling an object. We have also demonstrated an automated system that instantiates these design principles and can substantially reduce the effort required to produce good assembly instructions.

Our key insight is that *planning* a sequence of assembly operations that is easy to understand and *presenting* those operations in a clear and concise manner are strongly interrelated problems. Our system is based on this idea and considers both problems in parallel as it is designing the instructions.

Acknowledgements: We thank Boris Yamrom for his invaluable contributions to our system. Christina Vincent helped us run the psychology experiments. This work was supported by ONR grants N00014021034, N00014010717 and N00014001649.

References

80/20 Inc. 2003. www.8020.net.

Blanz, V., Tarr, M. J., and Bülthoff, H. H. 1999. What object attributes determine canonical views. *Perception 28*, 575–600.

Butz, A. 1997. Anymation with CATHI. In *Proceedings of AAAI/IAAI '97 in Providence / Rhode island*, AAAI Press, 957–962.

de Mello, L. S. H., and Sanderson, A. C. 1991. A correct and complete algorithm for the generation of mechanical assembly sequences. *IEEE Transactions on Robotics and Automation 7*, 2, 228–240.

Dresskill, E., and Cohen, E. 1995. Interactive design, analysis and illustration of assemblies. In *1995 Symposium on Interactive 3D Graphics*, ACM Press, 27–33.

Feiner, S. 1985. APEX An experiment in the automated creation of pictorial explanations. *IEEE Computer Graphics and Applications 5*, 11, 29–37.

Guibas, L. J., Halperin, D., Hirukawa, H., Latombe, J.-C., and Wilson, R. H. 1995. A simple and efficient procedure for polyhedral assembly partitioning under infinitesimal motions. In *IEEE International Conference on Robotics and Automation*, IEEE, 2553–2560.

Heiser, J., and Tversky, B. 2002. How to put things together. In *Poster presentation at the meeting of the Psychonomics Society*, Psychonomics Society.

Lin, M. C., and Canny, J. F. 1991. A fast algorithm for incremental distance calculation. In *IEEE Intern. Conf. on Robotics and Automation*, IEEE, 1008–1014.

Mackinlay, J. 1986. Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics 5*, 2, 110–141.

Martin, J. 1989. *High Tech Illustration*. North Light Books.

Mijksenaar, P., and Westendorp, P. 1999. *Open Here: The Art of Instructional Design*. Joost Elffers Books, New York.

Novick, L. R., and Morse, D. L. 2000. Folding a fish, making a mushroom: The role of diagrams in executing assembly procedures. *Memory & Cognition 28*, 1242–1256.

Palmer, S., Rosch, E., and Chase, P. 1981. Canonical perspective and the perception of objects. In *Attention and Performance IX*, 135–151.

Quinlan, S. 1994. Efficient distance computation between non-convex objects. In *IEEE Intern. Conf. on Robotics and Automation*, IEEE, 3324–3329.

Raab, A., and Rüger, M. 1996. 3D-ZOOM interactive visualization of structures and relations in complex graphics. In *3D Image Analysis and Synthesis*, 87–93.

Rist, T., Krüger, A., Schneider, G., and Zimmerman, D. 1994. AWA A workbench for semi-automated illustration design. In *Proc. of Advanced Visual Interfaces*, 59–68.

Romney, B., Godard, C., Goldwasser, M., and Ramkumar, G. 1995. An efficient system for geometric assembly sequence generation. *Proc. ASME International Computers in Engineering Conference*, 699–712.

Seligmann, D. D., and Feiner, S. 1991. Automated generation of intent-based 3D illustrations. In *Proceedings of SIGGRAPH 91*, 123–132.

Strothotte, T. 1998. *Computational Visualization. Graphics, Abstraction and Interactivity*. Springer, ch. 13, 215–240.

Tversky, B., and Hemenway, K. 1984. Objects, parts and categories. *Journal of Experimental Psychology: General 113*, 169–193.

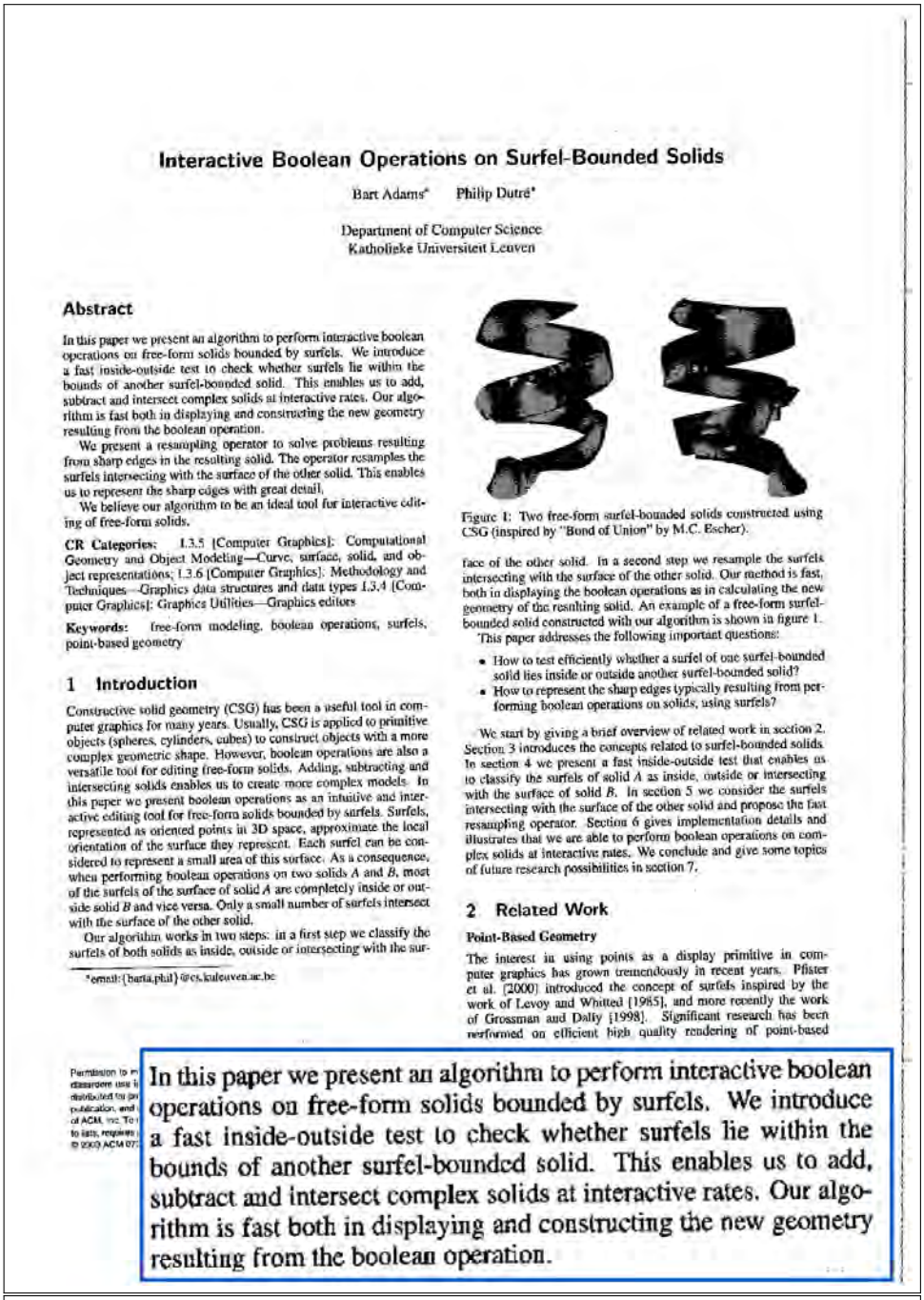
Tversky, B., Agrawala, M., Heiser, J., Lee, P., Hanrahan, P., Stolte, C., and Daniel, M.-P. Submitted. Cognitive design principles for automated generation of visualizations.

Wilson, R. H. 1992. *On Geometric Assembly Planning*. PhD thesis, Stanford University.

Wolter, J. D. 1989. On the automatic generation of assembly plans. In *Proc. IEEE International Conference on Robotics and Automation*, 62–68.

Zacks, J., Tversky, B., and Iyer, G. 2001. Perceiving, remembering and communicating structure in events. *Journal of Experimental Psychology: General 136*, 29–58.

FIGURE A.6 – Les deux premières et les deux dernières pages du troisième article scientifique



Interactive Boolean Operations on Surfel-Bounded Solids

Bart Adams* Philip Dutré*

Department of Computer Science
Katholieke Universiteit Leuven

Abstract

In this paper we present an algorithm to perform interactive boolean operations on free-form solids bounded by surfels. We introduce a fast inside-outside test to check whether surfels lie within the bounds of another surfel-bounded solid. This enables us to add, subtract and intersect complex solids at interactive rates. Our algorithm is fast both in displaying and constructing the new geometry resulting from the boolean operation.

We present a resampling operator to solve problems resulting from sharp edges in the resulting solid. The operator resamples the surfels intersecting with the surface of the other solid. This enables us to represent the sharp edges with great detail.

We believe our algorithm to be an ideal tool for interactive editing of free-form solids.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations; I.3.6 [Computer Graphics]: Methodology and Techniques—Graphics data structures and data types I.3.4 [Computer Graphics]: Graphics Utilities—Graphics editors

Keywords: free-form modeling, boolean operations, surfels, point-based geometry

1 Introduction

Constructive solid geometry (CSG) has been a useful tool in computer graphics for many years. Usually, CSG is applied to primitive objects (spheres, cylinders, cubes) to construct objects with a more complex geometric shape. However, boolean operations are also a versatile tool for editing free-form solids. Adding, subtracting and intersecting solids enables us to create more complex models. In this paper we present boolean operations as an intuitive and interactive editing tool for free-form solids bounded by surfels. Surfels, represented as oriented points in 3D space, approximate the local orientation of the surface they represent. Each surfel can be considered to represent a small area of this surface. As a consequence, when performing boolean operations on two solids A and B, most of the surfels of the surface of solid A are completely inside or outside solid B and vice versa. Only a small number of surfels intersect with the surface of the other solid.

Our algorithm works in two steps: in a first step we classify the surfels of both solids as inside, outside or intersecting with the surface of the other solid. In a second step we resample the surfels intersecting with the surface of the other solid. Our method is fast, both in displaying the boolean operations as in calculating the new geometry of the resulting solid. An example of a free-form surfel-bounded solid constructed with our algorithm is shown in figure 1.

*email:{barta,phil}@cs.kuleuven.ac.be

Figure 1: Two free-form surfel-bounded solids constructed using CSG (inspired by "Bond of Union" by M.C. Escher).

This paper addresses the following important questions:

- How to test efficiently whether a surfel of one surfel-bounded solid lies inside or outside another surfel-bounded solid?
- How to represent the sharp edges typically resulting from performing boolean operations on solids, using surfels?

We start by giving a brief overview of related work in section 2. Section 3 introduces the concepts related to surfel-bounded solids. In section 4 we present a fast inside-outside test that enables us to classify the surfels of solid A as inside, outside or intersecting with the surface of solid B. In section 5 we consider the surfels intersecting with the surface of the other solid and propose the fast resampling operator. Section 6 gives implementation details and illustrates that we are able to perform boolean operations on complex solids at interactive rates. We conclude and give some topics of future research possibilities in section 7.

2 Related Work

Point-Based Geometry

The interest in using points as a display primitive in computer graphics has grown tremendously in recent years. Pfister et al. [2000] introduced the concept of surfels inspired by the work of Levoy and Whitted [1985], and more recently the work of Grossman and Dally [1998]. Significant research has been performed on efficient high quality rendering of point-based

In this paper we present an algorithm to perform interactive boolean operations on free-form solids bounded by surfels. We introduce a fast inside-outside test to check whether surfels lie within the bounds of another surfel-bounded solid. This enables us to add, subtract and intersect complex solids at interactive rates. Our algorithm is fast both in displaying and constructing the new geometry resulting from the boolean operation.

In this paper we present an algorithm to perform interactive boolean operations on free-form solids bounded by surfels. We introduce a fast inside-outside test to check whether surfels lie within the bounds of another surfel-bounded solid. This enables us to add, subtract and intersect complex solids at interactive rates. Our algorithm is fast both in displaying and constructing the new geometry resulting from the boolean operation.

FIGURE A.7 – Différence entre l'image numérisée en haut et l'image d'origine vectorielle en bas

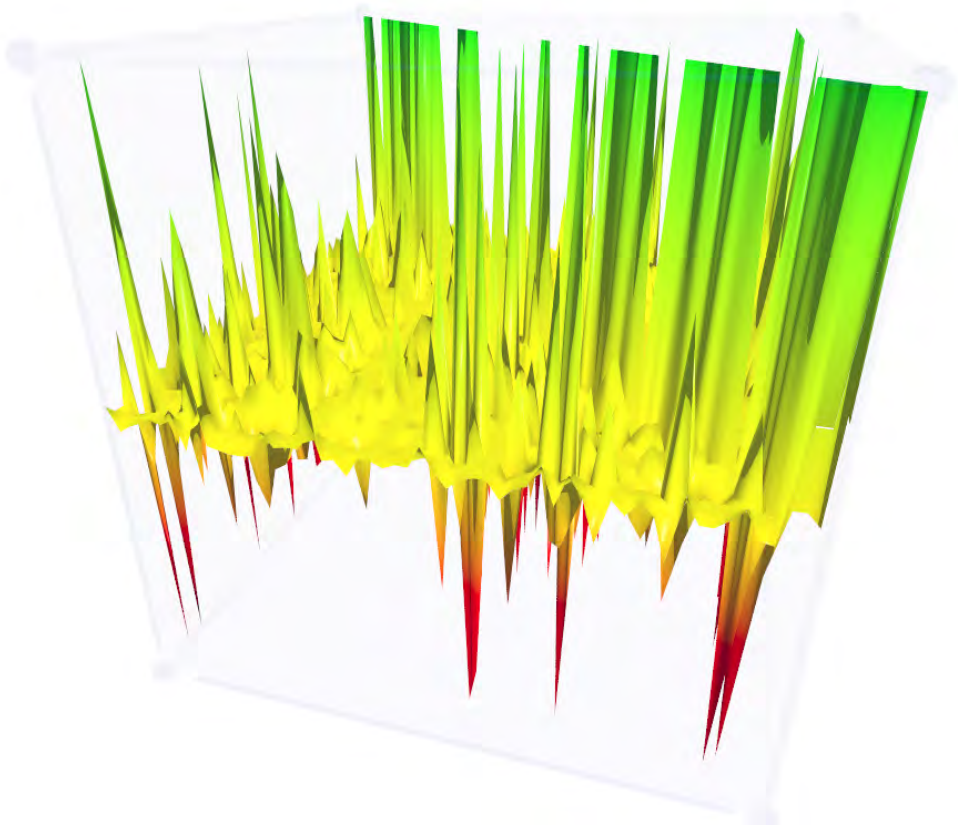


FIGURE A.8 – Vue tridimensionnelle de la matrice de covariance de la base d'apprentissage des articles sientifiques

Annexe B

La base MNIST

La base de données MNIST[12] est constituée d'images de chiffres manuscrits arabes. Elle se compose de deux parties, la première contient 60 000 échantillons destinés à l'apprentissage, la deuxième contient 10 000 autres échantillons pour effectuer le test. Les images de chiffres sont normalisées en taille et centrées. Toutes les images proviennent de la base NIST, à l'origine en noir et blanc, qui ont été réduites pour donner les images de dimension 28×28 (Fig. B.1,B.2) qui forment la base MNIST.

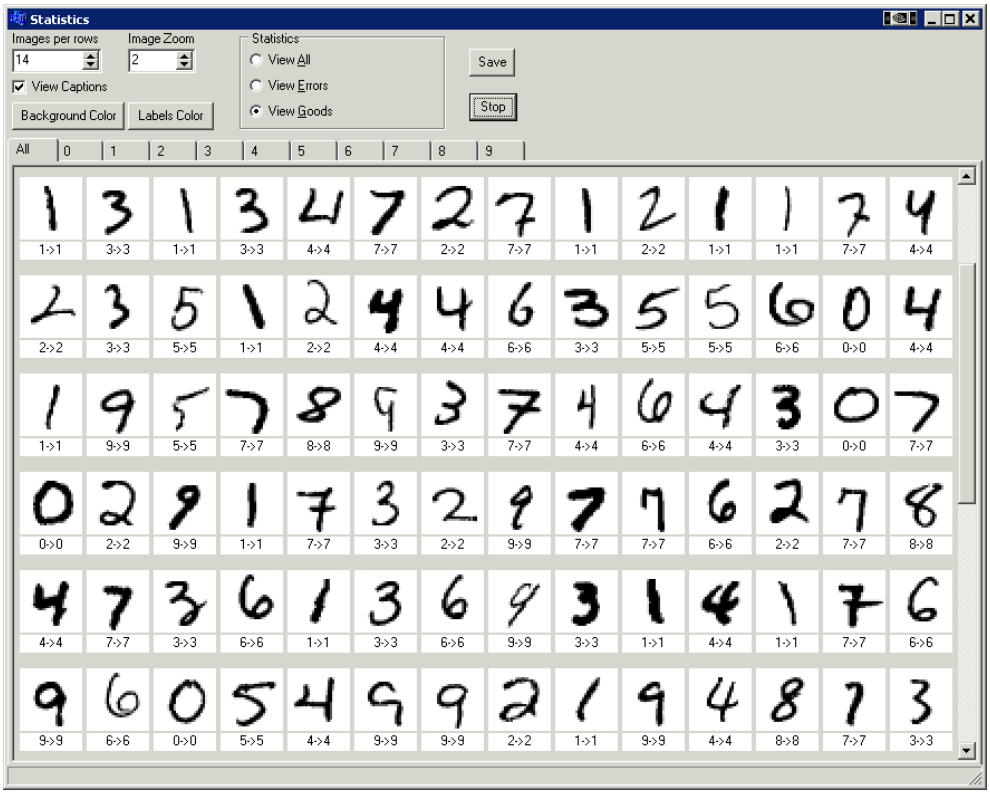


FIGURE B.1 – Exemple d'échantillons de la base MNIST reconnus par un PMC

[12] `http://yann.lecun.com/exdb/mnist/`

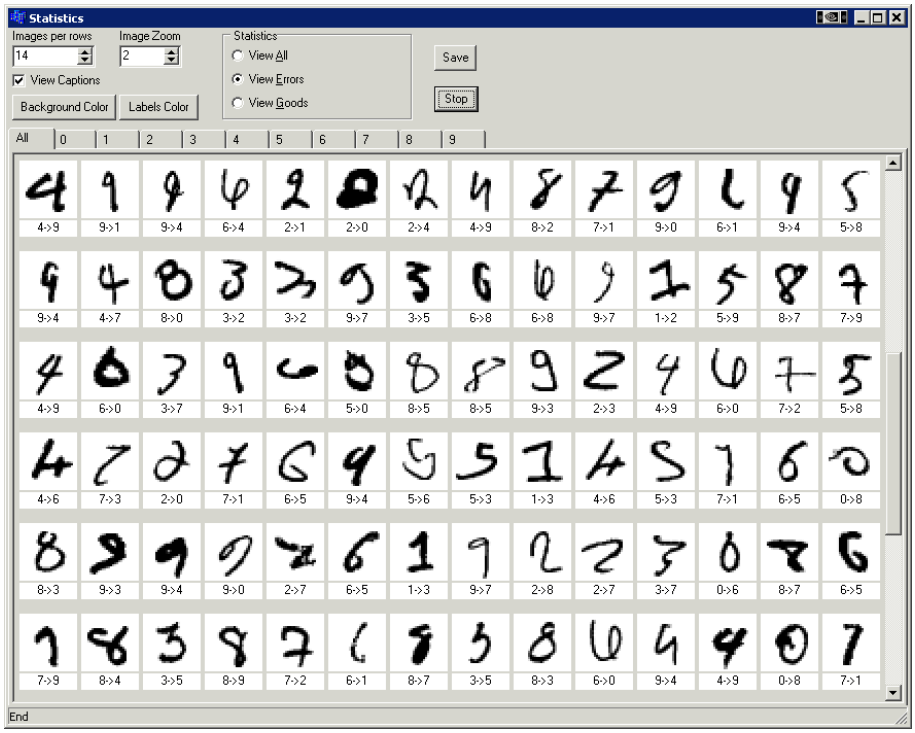
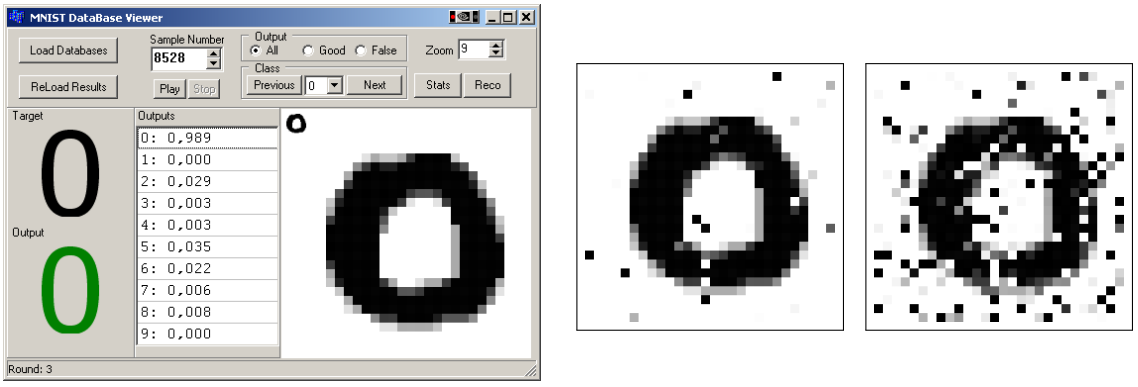


FIGURE B.2 – Exemple d'échantillons de la base MNIST non reconnus par un PMC



(a) (b) (c)

FIGURE B.3 – Exemple de création d'un représentant d'une classe par un algorithme génétique et utilisation d'un PMC. L'image (a) est le meilleur représentant actuel pour le zéro parmi les échantillons de la base de test, sa distance euclidienne avec le vecteur parfait est d'environ 0,05. L'image (b) est le résultat d'une simulation de la recherche du meilleur échantillon pour le chiffre un, en partant de l'image du meilleur zéro, sa distance avec le vecteur parfait est 10^{-2}. L'image (c) présente le même cas qu'en (b) mais avec une distance de 10^{-4}. On peut donc assez facilement tromper le PMC en maquillant un chiffre pour le faire passer pour un autre

Le tableau B.1 de [Le Cun et coll., 1998] donne une vue d'ensemble des résultats obtenus par différents classifieurs neuronaux, avec ou sans prétraitements de l'image.

Classifieur	Prétraitements	Erreur %
linear classifier (1-layer NN)	none	12
linear classifier (1-layer NN)	deskewing	8.4
pairwise linear classifier	deskewing	7.6
K-nearest-neighbors, Euclidean (L2)	none	5
K-nearest-neighbors, Euclidean (L2)	none	3.09
K-nearest-neighbors, L3	none	2.83
K-nearest-neighbors, Euclidean (L2)	deskewing	2.4
K-nearest-neighbors, Euclidean (L2)	deskewing, noise rem., blurring	1.8
K-nearest-neighbors, L3	deskewing, noise rem., blurring	1.73
K-nearest-neighbors, L3	deskewing, noise removal, blurring, 1 px shift	1.33
K-nearest-neighbors, L3	deskewing, noise removal, blurring, 2 px shift	1.22
K-NN, shape context matching	s.c. feature extraction	0.63
40 PCA + quadratic classifier	none	3.3
1000 RBF + linear classifier	none	3.6
K-NN, Tangent Distance	subsampling to 16x16	1.1
SVM, Gaussian Kernel	none	1.4
SVM deg 4 polynomial	deskewing	1.1
Reduced Set SVM deg 5 polynomial	deskewing	1
Virtual SVM deg-9 poly [distortions]	none	0.8
Virtual SVM, deg-9 poly, 1-pixel jittered	none	0.68
Virtual SVM, deg-9 poly, 1-pixel jittered	deskewing	0.68
Virtual SVM, deg-9 poly, 2-pixel jittered	deskewing	0.56
2-layer NN, 300 hidden units, mean square error	none	4.7
2-layer NN, 300 HU, MSE, [distortions]	none	3.6
2-layer NN, 300 HU	deskewing	1.6
2-layer NN, 1000 hidden units	none	4.5
2-layer NN, 1000 HU, [distortions]	none	3.8
3-layer NN, 300+100 hidden units	none	3.05
3-layer NN, 300+100 HU [distortions]	none	2.5
3-layer NN, 500+150 hidden units	none	2.95
3-layer NN, 500+300 HU, softmax, X-entropy, weight decay	none	1.53
2-layer NN, 800 HU, Cross-Entropy Loss	none	1.6
2-layer NN, 800 HU, cross-entropy [affine distortions]	none	1.1
2-layer NN, 800 HU, MSE [elastic distortions]	none	0.9
2-layer NN, 800 HU, cross-entropy [elastic distortions]	none	0.7
Convolutional net LeNet-1	subsampling to 16x16	1.7
Convolutional net LeNet-4	none	1.1
Convolutional net LeNet-4 with K-NN instead of last layer	none	1.1
Convolutional net LeNet-4 with local learning instead of ll	none	1.1
Convolutional net LeNet-5, [no distortions]	none	0.95
Convolutional net LeNet-5, [huge distortions]	none	0.85
Convolutional net LeNet-5, [distortions]	none	0.8
Convolutional net Boosted LeNet-4, [distortions]	none	0.7
Convolutional net, cross-entropy [affine distortions]	none	0.6
Convolutional net, cross-entropy [elastic distortions]	none	0.4

TABLEAU B.1 – Résultats obtenus sur la base MNIST [Le Cun et coll., 1998] pour différents classifieurs

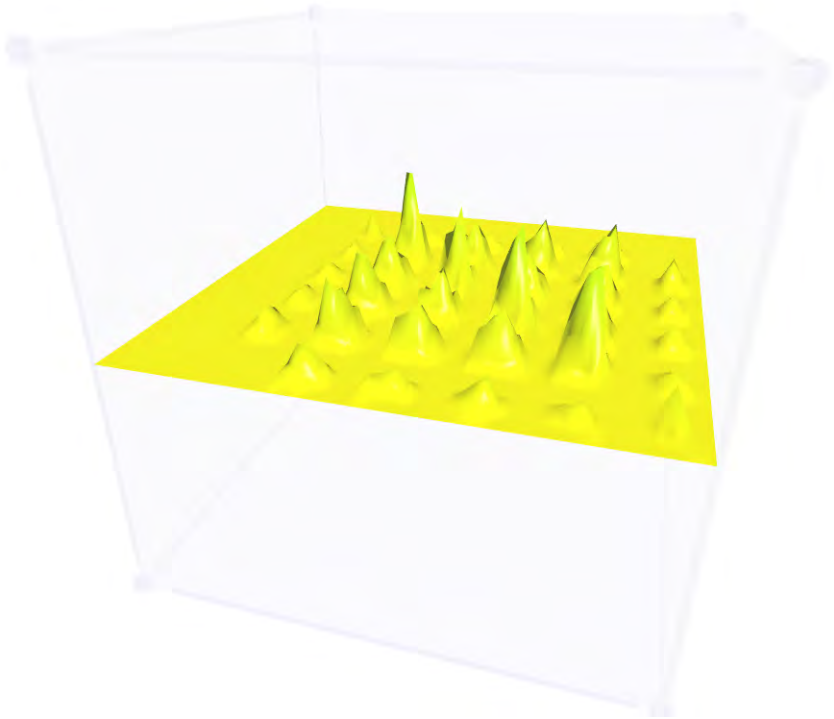


FIGURE B.4 – Vue tridimensionnelle de la matrice de covariance de la base d'apprentissage de la base MNIST réduite à des images de 7×7 pixels

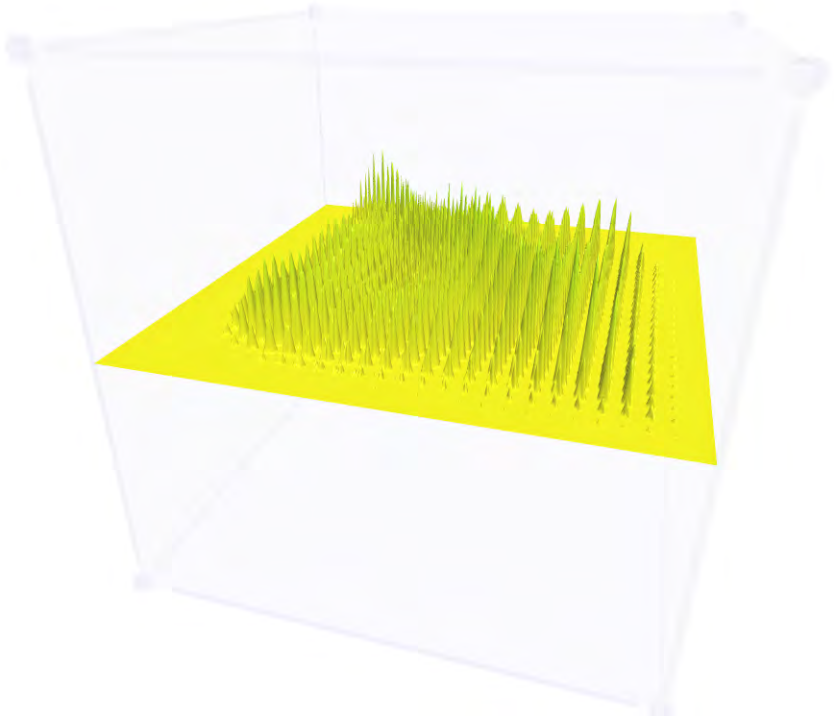


FIGURE B.5 – Vue tridimensionnelle de la matrice de covariance de la base d'apprentissage de la base MNIST composées d'images de 28×28 pixels

Annexe C

Perceptron multicouche

Le neurone biologique est une cellule nerveuse qui comprend un corps cellulaire (noyau et protoplasme), un prolongement axial cylindrique (axone) et des prolongements secondaires (dendrites). Les neurones reçoivent et émettent des signaux excitateurs ou inhibiteurs. Ces signaux sont transmis le long des axones puis atteignent les dendrites d'un autre neurone par l'intermédiaire de neurotransmetteurs [Kandel et coll., 2000]. Il existe près de dix mille types différents de neurones, le cerveau humain en comporterait environ cent milliards et chaque neurone est relié à dix mille autres en moyenne. Devant ces chiffres, il est plus facile de comprendre pourquoi le cerveau humain est robuste et tolérant aux fautes, flexible et facilement adaptable. Il s'accommode d'informations incomplètes, incertaines, ou bruitées et il est capable d'apprentissage. Toutes ces caractéristiques font de lui un modèle très séduisant ; c'est pourquoi, de nombreux systèmes informatiques de reconnaissance, actuels ou anciens, s'en inspirent.

Les neurones formels que nous utilisons tentent de reproduire les comportements des neurones biologiques. Ils s'en approchent dans la plupart des cas assez bien lorsqu'ils sont utilisés séparément. Le neurone informatique élémentaire peut être vu comme un automate qui reçoit des impulsions de ses autres neurones voisins, qui se charge à son tour de modifier cette information et de la renvoyer à ses voisins. La transmission de l'information se fait biologiquement par l'intermédiaire d'axones et de synapses qui sont simplement modélisés dans le modèle informatique par des paramètres scalaires (ou poids synaptiques) servant à pondérer les informations.

Le précurseur de la majorité des travaux sur les réseaux de neurones est le Perceptron de Rosenblatt [Rosenblatt, 1958]. Souvent à la base des réseaux statiques, le neurone formel en tant qu'abstraction du neurone physiologique a été proposé par [McCulloch et Pitts, 1943].

C.1 Rétropropagation du gradient de l'erreur

Il faut minimiser $E(x) = \sum_{p=1}^{P} E_p(w)$ avec $E_p(w) = \frac{1}{2} \sum_{q=1}^{N_L} (o_{L,q}(x_p) - d_q(x_p))^2$

Si l'on applique la descente de gradient à la variable w, il apparaît pour chaque poids $w_{l,j,i}$,

$$\begin{aligned}
w_{l,j,i} \leftarrow w_{l,i,j} &- \mu \frac{\partial E(w)}{\partial w_{l,j,i}} \\
w_{l,i,j} &- \mu \sum_{p=1}^{P} \frac{\partial E_p(w)}{\partial w_{l,j,i}} \\
w_{l,i,j} &- \mu \sum_{p=1}^{P} \frac{\partial E_p(w)}{\partial o_{l,j}} \frac{\partial o_{l,j}}{\partial w_{l,j,i}} \\
w_{l,i,j} &- \mu \sum_{p=1}^{P} \frac{\partial E_p(w)}{\partial o_{l,j}} \frac{\partial}{\partial w_{l,j,i}} f\left(\sum_{m=0}^{N_{l-1}} w_{l,j,m} o_{l-1,m} \right) \\
w_{l,i,j} &- \mu \sum_{p=1}^{P} \frac{\partial E_p(w)}{\partial o_{l,j}} f'\left(\sum_{m=0}^{N_{l-1}} w_{l,j,m} o_{l-1,m} \right) \frac{\partial}{\partial w_{l,j,i}} \left(\sum_{m=0}^{N_{l-1}} w_{l,j,m} o_{l-1,m} \right)
\end{aligned}$$

Comme $\forall m \neq i$, $\frac{\partial}{\partial w_{l,i,j}} \left(\sum_{m=0}^{N_{l-1}} w_{l,j,m} o_{l-1,m} \right) = 0$, il en résulte :

$$w_{l,j,i} \leftarrow w_{l,i,j} - \mu \sum_{p=1}^{P} \frac{\partial E_p(w)}{\partial o_{l,j}} f'\left(\sum_{m=0}^{N_{l-1}} w_{l,j,m} o_{l-1,m} \right) o_{l-1,i}$$

Si f est la sigmoïde, on a la propriété suivante :

$$\begin{aligned}
\frac{\partial f(x)}{\partial x} &= \frac{\partial}{\partial x}(1+e^{-x})^{-1} = -(-e^{-x})(1+e^{-x})^{-2} \\
&= f(x)e^{-x}(1+e^{-x})^{-1} = f(x)(1+e^{-x}-1)(1+e^{-x})^{-1} \\
&= f(x)(1-f(x))
\end{aligned}$$

D'où :

$$w_{l,i,j} \leftarrow w_{l,i,j} - \mu \sum_{p=1}^{P} \frac{\partial E_p(w)}{\partial o_{l,j}} o_{l,j}(1-o_{l,j}) o_{l-1,i}$$

Le terme $\frac{\partial E_p(w)}{\partial o_{l,j}}$ peut être exprimé en fonction des neurones des couches suivantes :

$$\begin{aligned}
\frac{\partial E_p(w)}{\partial o_{l,j}} &= \sum_{m=1}^{N_{l+1}} \frac{\partial E_p(w)}{\partial o_{l+1,m}} \frac{\partial o_{l+1,m}}{\partial o_{l,j}} \\
&= \sum_{m=1}^{N_{l+1}} \frac{\partial E_p(w)}{\partial o_{l+1,m}} \frac{\partial}{\partial o_{l,j}} f\left(\sum_{q=0}^{N_l} w_{l+1,m,q} o_{l,q} \right) \\
&= \sum_{m=1}^{N_{l+1}} \frac{\partial E_p(w)}{\partial o_{l+1,m}} o_{l+1,m}(1-o_{l+1,m}) w_{l+1,m,j}
\end{aligned}$$

Le terme $\frac{\partial E_p(w)}{\partial o_{l+1,m}}$ peut lui aussi être calculé de la même manière et continue ainsi jusqu'à parvenir à la couche de sortie où, cette fois-ci, le résultat est immédiat :

$$\frac{\partial E_p(w)}{\partial o_{L,j}} = o_{L,j}(x_p) - d_j(x_p) \tag{C.1}$$

Grâce à cette erreur de sortie connue, on peut remonter les liens et corriger chaque poids jusqu'à la couche d'entrée.

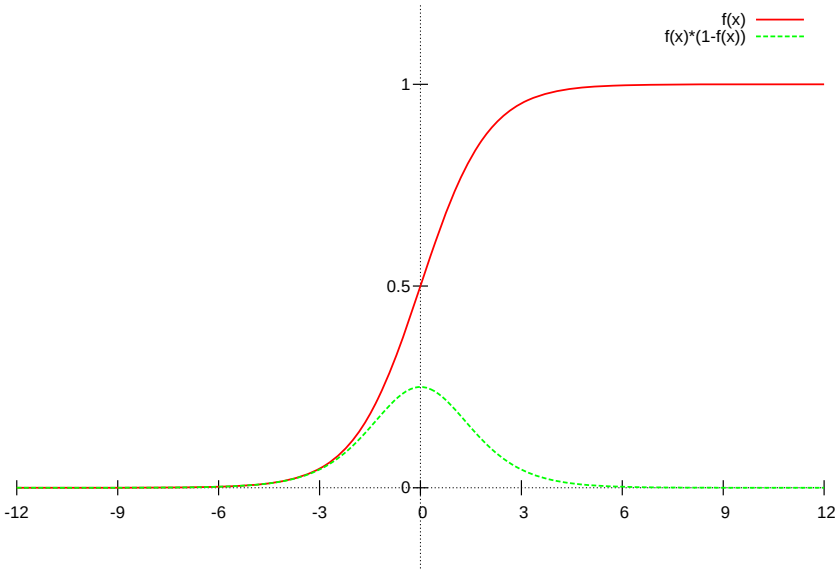


FIGURE C.1 – La sigmoïde et sa dérivée

C.2 Propriétés mathématiques

Les propriétés suivantes donnent un aperçu formel de la puissance des PMC. Les trois premières concernent uniquement un Perceptron seul, les suivantes sont relatives aux PMC :

- un Perceptron seul, linéaire à seuil à n entrées, divise l'espace des entrées $\mathbb{R}^n$ en deux sous-espaces délimités par un hyperplan. Réciproquement, tout ensemble linéairement séparable peut être discriminé par un Perceptron ;
- toute fonction booléenne linéairement séparable sur n variables peut être implantée par un Perceptron dont les poids synaptiques entiers w_i sont tels que $\lceil w_i \rceil \leqslant (n+1)^{n+\frac{1}{2}}$;
- il existe des fonctions booléennes linéairement séparables sur n variables qui requièrent des poids entiers supérieurs à $2^{n+\frac{1}{2}}$;
- toute fonction booléenne peut être calculée par un PMC linéaire à seuil comprenant une seule couche cachée ;
- deux couches cachées suffisent à représenter des frontières de décision convexes ;
- deux couches cachées suffisent à former une approximation arbitrairement proche de n'importe quelle surface de décision ou fonction continue non linéaire ;

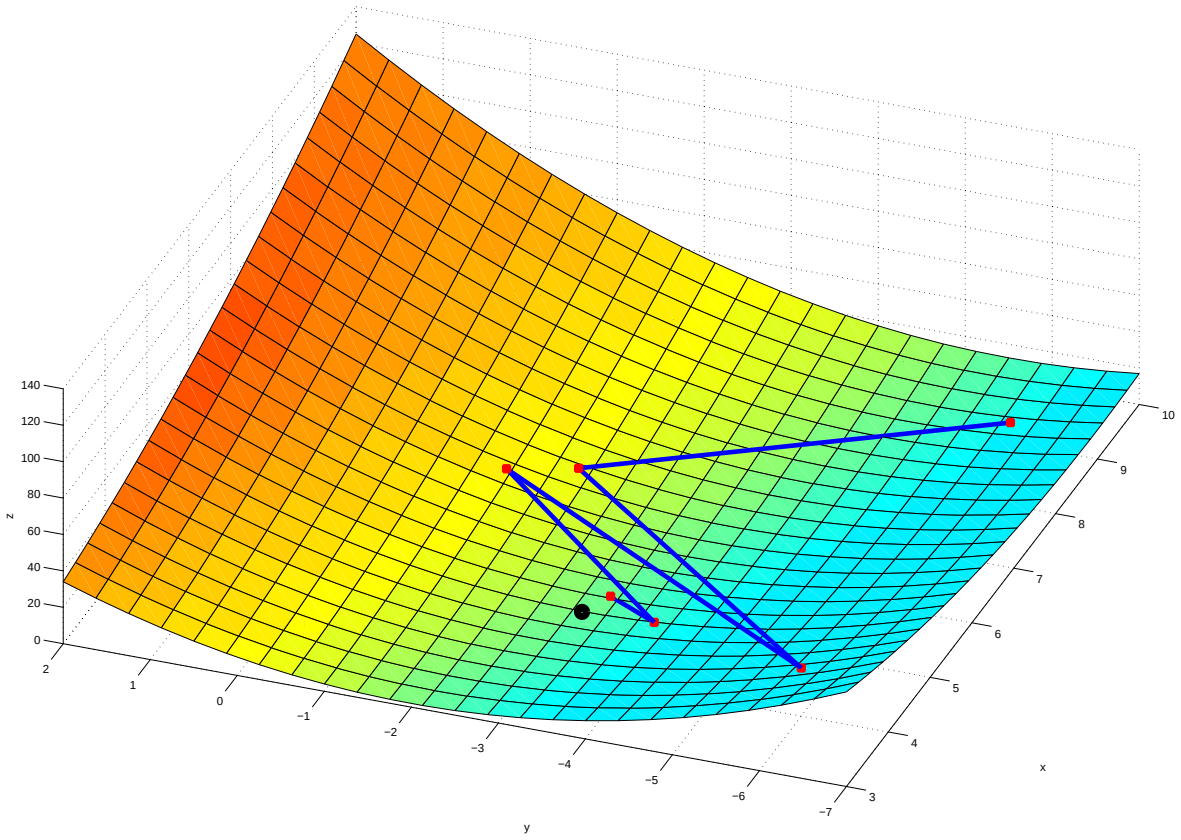


FIGURE C.2 – Exemple de descente de gradient sur une quadrique. La surface a pour équation $z = x^2 - 4x + 2xy + 2y^2 + 2y + 14$ et admet un minimum 1 en (5 ; -3). Le premier point pour effectuer la descente se trouve en (9 ; -6). En utilisant la fonction $x \rightarrow (1+i)^{-1}$ pour diminuer le pas d'apprentissage, où i est le numéro de l'itération, on obtient (5.3 ; -3.1) comme solution au bout de la cinquième itération. À l'itération 10, l'erreur d'approximation est de $2{\cdot}10^{-2}$. À l'itération 100, l'erreur est de $6{\cdot}10^{-4}$

- toutes les fonctions continues bornées sont représentables, avec une précision arbitraire, par un réseau à une seule couche cachée ;
- la plupart des fonctions numériques peuvent être approchées avec une précision arbitraire par des réseaux à une seule couche cachée ;
- il est possible d'implémenter toute bijection avec deux couches cachées si elles ont suffisamment d'éléments ;
- certains problèmes demandant un nombre exponentiel de neurones avec deux couches cachées et n'en demandent qu'un nombre polynomial si trois couches cachées sont utilisées ;
- le nombre d'exemples nécessaires à l'apprentissage d'un PMC est au pire de l'ordre de $n \log(n)$ si la fonction à approcher est booléenne de $\{0,1\}^n \rightarrow \{0,1\}$;
- si les exemples sont linéairement séparables, alors l'algorithme du Perceptron trouve la solution en un nombre fini d'itérations.

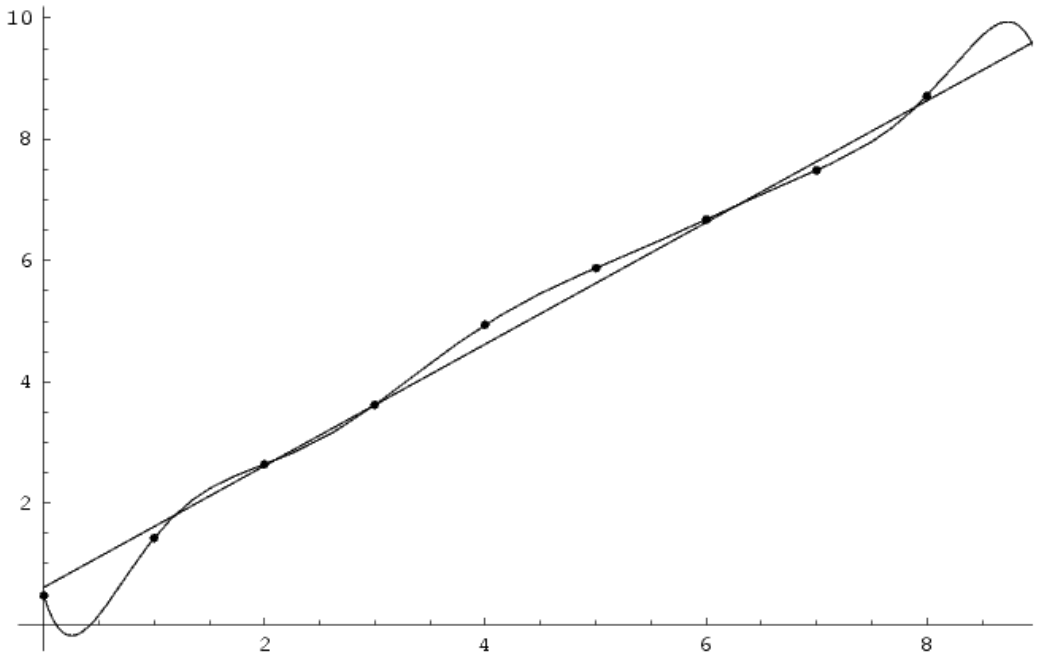


FIGURE C.3 – Exemple de surapprentissage : la fonction linéaire et polynomiale approchent correctement les données présentes dans le plan. Bien que le polynôme passe par tous les points et la droite par quelques-uns, cette dernière est bien meilleure pour une généralisation car elle a moins d'excursions aux extrémités

Annexe D

Réseaux récurrents statiques

D.1 Descente de gradient

L'équation à résoudre est :

$$\tau_i \frac{\partial V_i}{\partial t} = -V_i + f\left(\sum_{j=1}^{N} w_{ji} V_j\right) + x_i \tag{D.1}$$

En minimisant l'erreur quadratique par une descente de gradient, on obtient comme variation des poids :

$$\begin{aligned}\Delta w_{ij} &= -\eta \frac{\partial E}{\partial w_{ij}} = -\eta \frac{1}{2} \sum_k \frac{\partial E_k^2}{\partial w_{ij}} \\ &= -\eta \sum_k E_k \frac{\partial E_k}{\partial w_{ij}}\end{aligned}$$

Or $E_k = d_k - V_k$ donc :

$$\Delta w_{ij} = \eta \sum_k E_k \frac{\partial V_k}{\partial w_{ij}}$$

Le terme $\frac{\partial V_k}{\partial w_{ij}}$ peut être développé de la façon suivante :

$$\begin{aligned}\frac{\partial V_k}{\partial w_{ij}} &= \frac{\partial f(s_k)}{\partial w_{ij}} = \frac{\partial s_k}{\partial w_{ij}} f'(s_k) \\ &= f'(s_k) \frac{\partial}{\partial w_{ij}} \left(\sum_{q=1}^{N} w_{q_k} V_q\right) \\ &= f'(s_k) \left(\sum_{q=1}^{N} \frac{\partial}{\partial w_{ij}} (w_{q_k} V_q)\right) \\ &= f'(s_k) \sum_{q=1}^{N} \left(\frac{\partial}{\partial w_{ij}} w_{q_k} V_q + w_{q_k} \frac{\partial}{\partial w_{ij}} V_q\right)\end{aligned}$$

Comme $\frac{\partial w_{qk}}{\partial w_{ij}} = 1$ si $q = i$ et $k = j$, en utilisant le symbole de Kronecker, on obtient :

$$\begin{aligned}
\frac{\partial V_k}{\partial w_{ij}} &= f'(s_k)\left(\delta_{kj}V_i + \sum_q w_{qk}\frac{\partial V_q}{\partial w_{ij}}\right) \\
\iff \frac{\partial V_k}{\partial w_{ij}} - f'(s_k)\sum_q w_{qk}\frac{\partial V_q}{\partial w_{ij}} &= f'(s_k)\delta_{kj}V_i \\
\iff \sum_q \delta_{kq}\frac{\partial V_q}{\partial w_{ij}} - f'(s_k)\sum_q w_{qk}\frac{\partial V_q}{\partial w_{ij}} &= f'(s_k)\delta_{kj}V_i \\
\iff \sum_q \frac{\partial V_q}{\partial w_{ij}}\left(\delta_{kq} - f'(s_k)w_{qk}\right) &= f'(s_k)\delta_{kj}V_i \\
\iff \sum_q \frac{\partial V_q}{\partial w_{ij}}L_{kq} &= f'(s_k)\delta_{kj}V_i
\end{aligned}$$

avec $L_{kq} = \delta_{kq} - f'(s_k)w_{qk}$ d'où :

$$\frac{\partial V_q}{\partial w_{ij}}L_{kq} = (L^{-1})_{qj}f'(s_j)V_i \tag{D.2}$$

En reprenant l'expression de Δw_{ij} on a :

$$\Delta w_{ij} = \eta\sum_k E_k\frac{\partial V_k}{\partial w_{ij}} = \eta\sum_k E_k(L^{-1})_{kj}f'(s_j)V_i \tag{D.3}$$

Cette dernière équation est similaire à celle du cas statique. La difficulté ici est l'inversion de L : il faut connaître à l'avance tous les poids du réseau et, même dans le cas d'un calcul local, la complexité de l'inversion est $\mathcal{O}(N^3)$. Il est cependant possible de contourner le problème en introduisant une autre inversion : soit $X_j = \sum_k E_k(L^{-1})_{kj}f'(s_j)$, la règle de variation des poids devient $\Delta w_{ij} = \eta X_j V_i$ on définit un Y_j tel que :

$$X_j = f'(s_j)Y_j \tag{D.4}$$

avec

$$Y_j = \sum_k E_k(L^{-1})_{kj} \tag{D.5}$$

Une autre inversion donne :

$$\sum_k L_{kq}Y_j = E_q \tag{D.6}$$

On obtient alors :

$$Y_q = \sum_k f'(s_k)w_{kq}Y_k + E_q \tag{D.7}$$

qui est similaire à l'équation D.1 du début de section. On peut résoudre cette dernière équation en utilisant la même technique, en relaxant un réseau adjoint du système dynamique :

$$\tau\frac{\partial Y_q}{\partial dt} = -Y_q\sum_j f'(s_j)w_{jq}Y_j + E_q \tag{D.8}$$

Le réseau est analogue au réseau d'origine avec toutefois une substitution des poids w_{ij} entre i et j par des poids entre j et i de valeur $f'(s_i)w_{l,j,i}$, les neurones sont linéaires ($f(s) = s$) et l'erreur du neurone i devient la $i^{\text{ème}}$ entrée de l'adjoint.

La rétropropagation se résume donc à relaxer le réseau original jusqu'à un point fixe, comparer la sortie du réseau aux valeurs désirées de façon à obtenir les erreurs qui servent à alimenter le réseau adjoint, relaxer ce dernier afin d'avoir les Y_q et utiliser $\Delta w_{ij} = \eta \sum_k E_k (L^{-1})_{kj} f'(s_j) V_i$ pour trouver les poids d'origine.

Annexe E

Détermination des valeurs et vecteurs propres

Il existe un grand nombre de méthodes permettant de déterminer les vecteurs propres et les valeurs propres d'une matrice A.

Si A est carrée de dimensions $n \times n$, X un vecteur de dimension n et λ un scalaire, alors pour X non nul, les valeurs de λ vérifiant :

$$AX = \lambda X \tag{E.1}$$

sont appelées valeurs propres de la matrice A, les vecteurs correspondants étant les vecteurs propres.

Le passage entre l'équation mathématique et l'algorithmique n'est jamais évident car il pose des problèmes de faisabilité, de complexité et d'instabilité numériques. Certains algorithmes donnent à la fois les vecteurs et les valeurs propres, d'autres ne fournissent que les valeurs propres. Pour ces derniers, on utilise généralement la relation $(A - \lambda_i)I = 0$ où λ_i est la valeur propre trouvée et i la matrice identité. Une solution du système donnera donc le vecteur propre associé à λ_i et le processus sera réitéré pour tous les λ_i.

E.1 Méthodes du polynôme caractéristique

Les méthodes de Krylov ou Souriau déterminent le polynôme caractéristique $P_n(\lambda)$

$$P_n(\lambda) = \det(A - \lambda I) = \sum_i a_i \lambda^i \tag{E.2}$$

dont les n racines donnent les n valeurs propres de A. Une fois les coefficients trouvés, les racines sont trouvées par exemple par la méthode de Bairstow.

E.2 Méthode de la puissance itérée

La puissance itérée calcule la valeur propre de plus grand module et le vecteur propre qui lui est associé (Algo. 5). On utilise ensuite la méthode de déflation pour obtenir successivement les autres valeurs et vecteurs propres (Algo. 6).

$\lambda_0 \leftarrow 0$
V_0 un vecteur arbitraire
pour k *de* 1 *à* $+\infty$ **faire**
  $VP \leftarrow A \cdot V_0$
  $\lambda_1 \leftarrow max_i\{|VP_i|\}$
  $V_0 \leftarrow VP/\lambda_1$
  si $|\lambda_1 - \lambda_0| < \epsilon$ **alors**
    Stop
  sinon
    $\lambda_0 \leftarrow \lambda_1$
  fin
fin
V_0 est un vecteur propre associé à λ_0

ALGORITHME 5 – Méthode de la puissance itérée

Soit λ_0 la plus grande valeur propre
Soit V_0 le vecteur propre associé à λ_0
Soit Y_0 un vecteur propre de A^T
$A_1 = A - \lambda_0 V_0 Y_0^T / (Y_0^T V_0)$
A_1 a comme valeurs propres $0, \lambda_1, \ldots, \lambda_n$
Recommencer avec $A \leftarrow A_1$

ALGORITHME 6 – Méthode de la déflation

E.3 Méthodes des matrices semblables

L'idée des méthodes à suivre consiste à transformer la matrice A en une autre matrice qui rendra plus facile la détermination des valeurs et des vecteurs propres.

On dit que deux matrices A et B sont semblables si $\exists P, B = P^{-1}AP$. Si A et B sont semblables, alors elles ont les mêmes valeurs propres. Une autre propriété s'énonce : si λ valeur propre de A et V le vecteur propre associé, alors λ est aussi une valeur propre de B et Y est le vecteur propre correspondant avec $V = PY$.

E.3.1 Méthode de Crout

La méthode de Crout décompose une matrice A en deux matrices : L triangulaire inférieure à diagonale unité et R triangulaire supérieure telles que $A = LR$ (Algo. 7). L'algorithme est en $\mathcal{O}(n^3/3)$

pour *i et j de 1 à n* **faire**
  $l_{i,j} \leftarrow 0$
  $r_{i,j} \leftarrow 0$
fin
pour *j de 1 à n* **faire**
  $r_{1,j} \leftarrow a_{1,j}$
  $l_{j,j} \leftarrow 1$
  $l_{j,1} \leftarrow a_{j,1}/r_{1,1}$
fin
pour *i de 2 à n* **faire**
  pour *j de i à n* **faire**
    $r_{i,j} \leftarrow a_{i,j} - \sum_{k=1}^{i-1} l_{i,k} \cdot r_{k,j}$
  fin
  pour *j de $i+1$ à n* **faire**
    si $i < n$ **alors**
      $l_{j,i} \leftarrow (a_{j,i} - \sum_{k=1}^{i-1} l_{j,k} \cdot r_{k,i})/r_{i,i}$
    fin
  fin
fin

ALGORITHME 7 – Méthode de Crout

E.3.2 Méthode de Rutishauser

La méthode de Rutishauser utilise le fait que si $A = LR$ alors $A = LRLL^{-1}$. On a $B = RL$ semblable à A car $A = LBL^{-1}$. On peut donc utiliser les propriétés des matrices semblables pour déterminer les valeurs et vecteurs propres de A. L'algorithme de Rutishauser (Algo. 8) consiste à décomposer A par Crout en L_1R_1, de calculer $A_2 = RL$ et de décomposer à nouveau $A_2 = L_2R_2$ et ainsi de suite. La suite des A_i tend vers une matrice triangulaire dont les éléments diagonaux sont les valeurs propres de A

E.3.3 Méthode QR

La méthode QR utilise aussi l'idée de transformer la matrice A en une matrice semblable pour laquelle les calculs des valeurs propres est plus simple.

On se donne une matrice orthogonale Q_0 et $T_0 = Q_0^T A Q_0$, la méthode consiste à itérer l'algorithme 9 jusqu'à convergence.

Si A possède des valeurs propres réelles et distinctes en valeur absolue, alors la limite de T_k est une matrice triangulaire supérieure avec les valeurs propres de A sur la diagonale. Dans une implémentation «basique» de la méthode QR, la factorisation de T_{k-1} peut être effectuée en utilisant le procédé de Gram-Schmidt avec une complexité $\mathcal{O}(2n^3)$ mais avec une convergence assez lente.

La méthode est proche de celle de Rutishauser et plus souvent utilisée car la factorisation LR de Rutishauser perd en précision à chaque augmentation en module des coefficients sur-diagonaux de R.

répéter
  Crout donne L et R
  pour *i de 1 à n* **faire**
    pour *j de 1 à n* **faire**
      $A_{i,j} \leftarrow \sum_{k=1}^{n} R_{i,k} L_{k,j}$
    fin
  fin
  $ne \leftarrow 0$
  pour *i de 1 à n* **faire**
    si $|A_{i,i} - V_i| > \epsilon$ **alors**
      $ne \leftarrow ne + 1$
    fin
    $V_i = A_{i,i}$
  fin
jusqu'à $ne = 0$
Les valeurs propres sont dans les V_i

ALGORITHME 8 – Méthode de Rutishauser

Déterminer Q_k et R_k telles que
$Q_k R_k = T_{k-1}$ (factorisation QR)
$T_k = R_k Q_k$

ALGORITHME 9 – Itération k de la méthode QR

Des optimisations de la méthode QR sont possibles, on peut utiliser la variante QR-Hessenberg qui démarre la méthode avec une matrice T telle que $\forall i > j+1, t_{ij} = 0$. La complexité du calcul des T_k est alors de $\mathcal{O}(n^2)$. Pour gagner en précision et stabilité, on utilise généralement une réduction de A par la méthode de Householder et une factorisation de T_k par la méthode de Givens plutôt que le procédé de Gram-Schmidt. La convergence peut aussi être améliorée lorsque les valeurs propres sont proches les unes des autres par l'utilisation d'une translation (Algo. 10) ou bien encore par une méthode de double translation comme dans MATLAB pour s'assurer de la convergence des itérations QR.

Déterminer Q_k et R_k telles que
$Q_k R_k = T_{k-1} - \mu I$ (factorisation QR)
$T_k = R_k Q_k + \mu I$

ALGORITHME 10 – Itération k de la méthode QR avec translation

La méthode de Householder a été implémentée est donne de très bons résultats. Nous avons testé la robustesse sur des matrices de différentes tailles, la décomposition de A en $A' = V\Lambda V^T$ est utilisée pour évaluer $A - A'$ et la moyenne de tous les coefficients de cette matrice est utilisée comme score de précision. Pour des matrices 3×3 allant jusqu'à 400×400, l'erreur varie respectivement de 10^{-48} à 10^{-39} avec 50 itérations. À titre de comparaison, dans les mêmes conditions, l'algorithme de Crout donne déjà une erreur de 10^{-6} pour des matrices de taille 100×100 et dans des temps largement plus élevés.

Bibliographie

[Ahmed, 1995] P. Ahmed. A neural network based dedicated thinning method. *Pattern Recognition Letters*, 16(6):585–590, 1995.

[Aiello et coll., 2002] M. Aiello, C. Monz, L. Todoran et M. Worring. Document understanding for a broad class of documents. *International Journal on Document Analysis and Recognition*, 5(1):1–16, 2002.

[Akindele et Belaïd, 1993] O. T. Akindele et A. Belaïd. Page segmentation by segment tracing. *International Conference on Document Analysis and Recognition*, 1(2):341–344, 1993.

[Akindele et Belaïd, 1995] O. T. Akindele et A. Belaïd. Construction of generic models of document structures using inference of tree grammars. *International Conference on Document Analysis and Recognition*, 1(3):206–209, 1995.

[Alam et coll., 2003] H. Alam, R. Hartono, A. Kumar, A. F. R. Rahman, Y. Tarnikova et C. Wilcox. Assuming accurate layout information for web documents is available, what now? *International Workshop on Document Layout Interpretation and its Applications*, 1(3):27–30, 2003.

[Almeida, 1987] L. B. Almeida. Backpropagation in Perceptrons with feedback. *Neural Computers*, pages 199–208, 1987.

[Altamura et coll., 2001] O. Altamura, F. Esposito et D. Malerba. Transforming paper documents into XML format with Wisdom++. *International Journal on Document Analysis and Recognition*, 4(1):2–17, 2001.

[Amano et coll., 2004] A. Amano, N. Asada et M. Mukunoki. Modification table form generation system based on the form recognition. *International Conference on Pattern Recognition*, 2(17):659–662, 2004.

[Amin et coll., 1996] A. Amin, H. Al-Sadoun et S. Fischer. Hand-printed Arabic character recognition system using an artificial network. *Pattern Recognition*, 29(4):663–675, 1996.

[André et coll., 1989] J. André, R. Furuta et V. Quint. *Structured Documents*. Cambridge University Press, 1989.

[Anigbogu et Belaïd, 1995] J. C. Anigbogu et A. Belaïd. Hidden markov models in text recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 9(6):925–958, 1995.

[Anquetil, 1997] É. Anquetil. Modélisation et reconnaissance par la logique floue : application à la lecture automatique en-ligne de l'écriture manuscrite omni-scripteur. *Thèse de l'Université de Rennes 1*, 1997.

[Arthur et Vassilvitskii, 2007] D. Arthur et S. Vassilvitskii. K-means++: the advantages of careful seeding. *Symposium on Discrete Algorithms*, 1(18):1027–1035, 2007.

[Azzabou et Likforman-Sulem, 2004] N. Azzabou et L. Likforman-Sulem. Neural network-based proper names extraction in fax images. *International Conference on Pattern Recognition*, 2(17):421–424, 2004.

[Baccino et Colé, 1995] T. Baccino et P. Colé. *La Lecture Experte*. Presses Universitaires de France, 1995.

[Baird et coll., 1990] H. S. Baird, S. E. Jones et S. J. Fortune. Image segmentation by shape-directed covers. *International Conference on Pattern Recognition*, 1(10):820–825, 1990.

[Balci et Atalay, 2002] K. Balci et V. Atalay. PCA for gender estimation: which eigenvectors contribute? *International Conference on Pattern Recognition*, 3(16):363–366, 2002.

[Baldi, 1995] P. Baldi. Gradient descent learning algorithm overview: a general dynamical systems perspective. *IEEE Transactions on Neural Networks*, 6(1):182–195, 1995.

[Becker, 1976] C. A. Becker. Allocation of attention during visual word recognition. *Journal of Experimental Psychology: Human Perception and Performance*, 2(4):556–566, 1976.

[Belaïd, 1997] A. Belaïd. Analyse de documents: de l'image à la représentation par les normes de codage. *Revue Document Numérique*, 1(1):1–17, 1997.

[Belaïd et coll., 2005] A. Belaïd, A. Alusse, Y. Rangoni, H. Cecotti et coll. Document retro-conversion for personalized electronic reedition. *International Workshop on Document Analysis*, 1(1):193–218, 2005.

[Belaïd et coll., 1993] A. Belaïd, J. C. Anigbogu et Y. Chenevoy. Qualitative analysis of low-level logical structures. *Electronic Publishing*, 6(1):435–446, 1993.

[Belaïd et coll., 2000] A. Belaïd, L. Pierron et N. Valverde. Part-of-speech tagging for table of contents recognition. *International Conference on Pattern Recognition*, 4(15):451–454, 2000.

[Belaïd et coll., 2007] A. Belaïd, Y. Rangoni et I. Falk. XML data representation in document image analysis. *International Conference on Document Analysis and Recognition*, 1(9), 2007.

[Belaïd et Toussaint, 2000] A. Belaïd et Y. Toussaint. Une méthode d'étiquetage morpho-syntaxique pour la reconnaissance de tables de matières. *Colloque International Francophone sur l'Écrit et le Document*, 1(2):51–60, 2000.

[Belaïd et coll., 2004] A. Belaïd, I. Turcan, J.-M. Pierrel, Y. Belaïd, Y. Rangoni et H. Hadjamar. Automatic indexing and reformulation of ancient dictionaries. *International Conference on Document Image Analysis for Libraries*, 1(1):342–354, 2004.

[Belaïd et coll., 1998] Y. Belaïd, J.-L. Panchèvre et A. Belaïd. Form analysis by neural classification of cells. *International Workshop on Document Analysis Systems*, 1655(3):58–71, 1998.

[Besagni et coll., 2003] D. Besagni, A. Belaïd et N. Benet. A segmentation method for bibliographic references by contextual tagging of fields. *International Conference on Document Analysis and Recognition*, 1(7):384– 388, 2003.

[Bi et coll., 2003] J. Bi, K. P. Bennett, M. Embrechts, C. M. Breneman et M. Song. Dimensionality reduction via sparse support vector machines. *Journal of Machine Learning Research*, 3:1229–1243, 2003.

[Bloechle et coll., 2006] J.-L. Bloechle, M. Rigamonti, K. Hadjar, D. Lalanne et R. Ingold. XCDF: a canonical and structured document format. *International Workshop on Document Analysis Systems*, 3872(7):141–152, 2006.

[Blostein et coll., 1996] D. Blostein, H. Fahmy et A. Grbavec. Issues in the practical use of graph rewriting. *International Workshop on Graph Grammars and their Application to Computer Science*, 1073(5):38–55, 1996.

[Blum et Langley, 1997] A. Blum et P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271, 1997.

[Bouriel et coll., 2005] K. Bouriel, S. Snoussi Maddouri et K. Hamrouni. Un système neuronal pour la reconnaissance de mots arabes manuscrits. *Conférence Internationale sur les Sciences Électroniques, Technologies de l'Information et des Télécommunications*, 1(3), 2005.

[Bramall et Higgins, 1995] P. E. Bramall et C. A. Higgins. A cursive script-recognition system based on human reading models. *Machine Vision and Applications*, 8(4):224–231, 1995.

[Breiman, 1996] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[Breiman, 1998] L. Breiman. Arcing classifiers. *The Annals of Statistics*, 26(3):801–824, 1998.

[Breuel, 2003] T. M. Breuel. An algorithm for finding maximal whitespace rectangles at arbitrary orientations for document layout analysis. *International Conference on Document Analysis and Recognition*, 1(7):66–70, 2003.

[Bromley et coll., 1994] J. Bromley, I. Guyon, Y. Le Cun, E. Säckinger et R. Shah. Signature verification using a siamese time delay neural network. *Advances in Neural Information Processing Systems*, 6:737–744, 1994.

[Brown, 1989] B. Brown. Standards for structured documents. *The Computer Journal*, 32(6):505–514, 1989.

[Brugger et coll., 1998] R. Brugger, F. Bapst et R. Ingold. A DTD extension for document structure recognition. *International Conference on Electronic Publishing*, 1375(7):343–354, 1998.

[Brugger et coll., 1997] R. Brugger, A. Zramdini et R. Ingold. Modeling documents for structure recognition using generalized n-grams. *International Conference on Document Analysis and Recognition*, 1(4):56–60, 1997.

[Burge et Monagan, 1995] M. Burge et G. Monagan. Using the Voronoi tessellation for grouping words and multipart symbols in documents. *Vision Geomerty*, 2573(4):116–123, 1995.

[Carreira-Perpiñán, 1997] M. Á. Carreira-Perpiñán. A review of dimension reduction techniques. *Technical Report CS-96-09*, 1(1):1–69, 1997.

[Casasent et coll., 1997] D. Casasent, M. Frydrych, J. Parkkinen et A. Visa. New techniques for object detection and recognition. *Scandinavian Conference on Image Analysis*, 1(10):597–604, 1997.

[Casey et Lecolinet, 1996] R. G. Casey et É. Lecolinet. A survey of methods and strategies in character segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):690–706, 1996.

[Cattell, 1996] R. B. Cattell. The scree test for the number of factors. *Multivariate Behavioral Research*, 1(1):245–276, 1996.

[Cattoni et coll., 1998] R. Cattoni, T. Coianiz, S. Messelodi et C. M. Modena. Geometric layout analysis techniques for document image understanding: a review. *Technical Report, ITC-IRST, Italie*, pages 1–68, 1998.

[Ceci et coll., 2005] M. Ceci, M. Berardi et D. Malerba. Relational learning techniques for document image understanding: comparing statistical and logical approaches. *International Conference on Document Analysis and Recognition*, 1(8):473–482, 2005.

[Cecotti et Belaïd, 2005] H. Cecotti et A. Belaïd. Rejection strategy for convolutional neural network by adaptive topology applied to handwritten digits recognition. *International Conference on Document Analysis and Recognition*, 1(8):765–769, 2005.

[Ceheux, 2002] G. R. Ceheux. Stratégies pour l'interprétation de documents. *Actes des Deuxièmes Assises Nationale du GdR I3*, pages 257–288, 2002.

[Cesarini et coll., 2003] F. Cesarini, E. Francesconi, M. Gori et G. Soda. Analysis and understanding of multi-class invoices. *International Journal on Document Analysis and Recognition*, 6(2):102–114, 2003.

[Cesarini et coll., 2001] F. Cesarini, M. Lastri, S. Marinai et G. Soda. Encoding of modified X-Y trees for document classification. *International Conference on Document Analysis and Recognition*, 2(6):1131–1136, 2001.

[Chang et Chen, 2001] M.-T. Chang et S.-Y. Chen. Deformed trademark retrieval based on 2D pseudo-hidden Markov model. *Pattern Recognition*, 34(5):953–967, 2001.

[Chenevoy et Belaïd, 1991] Y. Chenevoy et A. Belaïd. Hypothesis management for structured document recognition. *International Conference on Document Analysis and Recognition*, 1(1):121–129, 1991.

[Chi et Wong, 2001] Z. Chi et K. W. Wong. A two-stage binarization approach for document images. *International Symposium on Intelligent Multimedia, Video and Speech Processing*, 1:275–278, 2001.

[Choisy et Belaïd, 2002] C. Choisy et A. Belaïd. Cross-learning in analytic word recognition without segmentation. *International Journal on Document Analysis and Recognition*, 4(4):281–286, 2002.

[Clarke et coll., 1995] C. L. A. Clarke, G. V. Cormack et F. J. Burkowski. An algebra for structured text search and a framework for its implementation. *The Computer Journal*, 38(1):43–56, 1995.

[Cohen et coll., 1994] E. Cohen, J. J. Hull et S. N. Srihari. Control structure for interpreting handwritten addresses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(10):1049–1055, 1994.

[Coltheart et Rastle, 1994] M. Coltheart et K. Rastle. Serial processing in reading aloud: evidence for dual-route models of reading. *Journal of Experimental Psychology: Human Perception and Performance*, 20(6):1197–1211, 1994.

[Conway, 1993] A. Conway. Page grammars and page parsing. A syntactic approach to document layout recognition. *International Conference on Document Analysis and Recognition*, 1(2):761–764, 1993.

[Cornuéjols et Miclet, 2002] A. Cornuéjols et L. Miclet. *Apprentissage Artificiel - Concepts et Algorithmes*. Eyrolles, 2002.

[Côté, 1997] M. Côté. Utilisation d'un modèle d'accès lexical et de concepts perceptifs pour la reconnaissance d'images de mots cursifs. *Thèse de l'École Nationale Supérieure des Télécommunications*, 1997.

[Côté et coll., 1998] M. Côté, É. Lecolinet, M. Cheriet et C. Y. Suen. Automatic reading of cursive scripts using a reading model and perceptual concepts. The Perceptro system. *International Journal on Document Analysis and Recognition*, 1(1):3–17, 1998.

[Coüasnon, 2006] B. Coüasnon. Dmos, a generic document recognition method: application to table structure analysis in a general and in a specific way. *International Journal on Document Analysis and Recognition*, 8(2-3):111–122, 2006.

[Datta et coll., 2001] A. Datta, S. K. Parui et B. B. Chaudhuri. Skeletonization by a topology adaptive self-organizing neural network. *Pattern Recognition*, 3(34):617–629, 2001.

[de Oliveira et coll., 2001] L. E. S. de Oliveira, N. Benahmed, R. Sabourin, F. Bortolozzi et C. Y. Suen. Feature subset selection using genetic algorithms for handwritten digit recognition. *Brazilian Symposium on Computer Graphics and Image Processing*, 1(14):362–369, 2001.

[de Oliveira et coll., 2006] L. E. S. de Oliveira, M. Morita et R. Sabourin. Feature selection for ensembles applied to handwriting recognition. *International Journal on Document Analysis and Recognition*, 8(4):262–279, 2006.

[Delalandre et coll., 2003] M. Delalandre, S. Nicolas, E. Trupin et J.-M. Ogier. Reconnaissance de symboles par approche structurelle globale-locale basée sur l'utilisation de scénarios et exploitant une représentation XML des données. *International Conference on Image and Signal Processing*, pages 631–639, 2003.

[Dengel et coll., 1992] A. R. Dengel, R. Bleisinger, R. Hoch, F. Fein et F. Hones. From paper to office document standard representation. *Computer*, 25(7):63–67, 1992.

[Dengel et Dubiel, 1996] A. R. Dengel et F. Dubiel. Computer understanding of document structure. *International Journal of Imaging Systems and Technology*, 7(4):271–278, 1996.

[Dengel et Klein, 2002] A. R. Dengel et B. Klein. SmartFIX: a requirements-driven system for document analysis and understanding. *International Workshop on Document Analysis Systems*, 2423(5):77–88, 2002.

[Derrien-Péden, 1991] D. Derrien-Péden. Frame-based system for macro-typographical structure analysis in scientific papers. *International Conference on Document Analysis and Recognition*, 1(1):311–319, 1991.

[Diligenti et coll., 2001] M. Diligenti, M. Gori, M. Maggini et E. Martinelli. Adaptive graphical pattern recognition for the classification of company logos. *Pattern Recognition*, 34(10):2049–2061, 2001.

[Doermann, 1998] D. S. Doermann. The indexing and retrieval of document images: a survey. *Computer Vision and Image Understanding*, 70(3):287–298, 1998.

[Doermann et coll., 1996] D. S. Doermann, E. Rivlin et I. Weiss. Applying algebraic and differential invariants for logo recognition. *Machine Vision and Applications*, 9(2):73–86, 1996.

[Dori et coll., 1995] D. Dori, D. S. Doermann, C. Shin, R. M. Haralick, I. Phillips, M. Buchman et D. Ross. The representation of document structure: a generic object-process analysis. *Computer Vision Laboratory Series*, 3521, 1995.

[Dreyfus et coll., 2002] G. Dreyfus, J. Martinez, M. Samuelides, M. Gordon, F. Badran, S. Thiria et L. Hérault. *Réseaux de Neurones - Méthodologie et Applications*. Eyrolles, 2002.

[Eastwood et coll., 1997] B. Eastwood, A. Jennings et A. Harvey. Neural network based segmentation of handwritten words. *International Conference on Image Processing and its Applications*, 2(6):750–755, 1997.

[Esposito et coll., 1992] F. Esposito, D. Malerba et G. Semeraro. Classification in noisy environments using a distance measure between structural symbolic descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(3):390–402, 1992.

[Esposito et coll., 2004] F. Esposito, D. Malerba, G. Semeraro, S. Ferilli, O. Altamura, T. M. A. Basile, M. Berardi, M. Ceci et N. Di Mauro. Machine learning methods for automatically processing historical documents: from paper acquisition to XML transformation. *International Conference on Document Image Analysis for Libraries*, 1(1):328–335, 2004.

[Etemad et Chellappa, 1997] K. Etemad et R. Chellappa. Discriminant analysis for recognition of human face images. *International Conference on Audio- and Video-Based Biometric Person Authentication*, 1206(1):127–142, 1997.

[Etemad et coll., 1994] K. Etemad, R. Chellappa et D. S. Doermann. Document page segmentation by integrating distributed soft decisions. *International Conference on Neural Networks*, 6:4022–4027, 1994.

[Etemad et coll., 1997] K. Etemad, D. S. Doermann et R. Chellappa. Multiscale segmentation of unstructured document pages using soft decision integration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1):92–96, 1997.

[Fahlman et Lebiere, 1990] S. E. Fahlman et C. Lebiere. The cascade-correlation learning architecture. *Advances in Neural Information Processing Systems*, 1(2):524–532, 1990.

[Fisher, 1991] J. L. Fisher. Logical structure descriptions of segmented document images. *International Conference on Document Analysis and Recognition*, 1(1):302–310, 1991.

[Fletcher et Kasturi, 1988] L. A. Fletcher et R. Kasturi. A robust algorithm for text string separation from mixed text/graphics images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):910–918, 1988.

[Francesconi et coll., 1997] E. Francesconi, P. Frasconi, M. Gori, S. Marinai, J. Q. Sheng, G. Soda et A. Sperduti. Logo recognition by recursive neural networks. *International Workshop on Graphics Recognition, Algorithms and Systems*, 1389(2):104–117, 1997.

[Frasconi et coll., 1998] P. Frasconi, M. Gori et A. Sperduti. A general framework for adaptive processing of data structures. *IEEE Transactions on Neural Networks*, 9(5):768–786, 1998.

[Freund, 1995] Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, 1995.

[Garain et coll., 2004a] U. Garain, B. B. Chaudhuri et A. R. Chaudhuri. Identification of embedded mathematical expressions in scanned documents. *International Conference on Pattern Recognition*, 1(17):384–387, 2004a.

[Garain et coll., 2004b] U. Garain, B. B. Chaudhuri et R. P. Ghosh. A multiple-classifier system for recognition of printed mathematical symbols. *International Conference on Pattern Recognition*, 1(17):380–383, 2004b.

[Gonzalez et coll., 1976] R. C. Gonzalez, J. J. Edwards et M. G. Thomason. An algorithm for the inference of tree grammars. *International Journal of Parallel Programming*, 5(2):145–164, 1976.

[Gori et coll., 2003] M. Gori, M. Maggini, S. Marinai, J. Q. Sheng et G. Soda. Edge-backpropagation for noisy logo recognition. *Pattern Recognition*, 36(1):103–110, 2003.

[Gyohten et coll., 1995] K. Gyohten, T. Sumiya, N. Babaguchi, K. Kakusho et T. Kitahashi. Extracting characters and character lines in multi-agent scheme. *International Conference on Document Analysis and Recognition*, 1(3):305–308, 1995.

[Hadjar et coll., 2002] K. Hadjar, O. Hitz, L. Robadey et R. Ingold. Configuration REcognition Model for Complex Reverse Engineering Methods: 2(CREM). *International Workshop on Document Analysis Systems*, 2423(5):523–530, 2002.

[Hall et Smith, 1997] M. A. Hall et L. A. Smith. Feature subset selection: a correlation based filter approach. *International Conference on Neural Information*, 1(4):855–858, 1997.

[Hamza et coll., 2005] H. Hamza, E. Smigiel et A. Belaïd. Neural based binarization techniques. *International Conference on Document Analysis and Recognition*, 1(8):317–321, 2005.

[Hancock et Wilson, 2002] E. R. Hancock et R. C. Wilson. Graph-based methods for vision: a Yorkist manifesto. *International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pages 31–46, 2002.

[Haralick, 1994] R. M. Haralick. Document image understanding: geometric and logical layout. *Conference on Computer Vision and Pattern Recognition*, pages 385–390, 1994.

[Hassibi et Stork, 1993] B. Hassibi et D. G. Stork. Second order derivatives for network pruning: optimal brain surgeon. *Advances in Neural Information Processing Systems*, 5:164–171, 1993.

[Hertz et coll., 1991] J. Hertz, A. Krogh et R. G. Palmer. *An Introduction to the Theory of Neural Computation*. Addison-Wesley, 1991.

[Hopfield, 1982] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8):2554–2558, 1982.

[Héroux et coll., 2000] P. Héroux, E. Trupin et Y. Lecoutier. Structural classification for retrospective conversion of documents. *Advances in Pattern Recognition*, 1876(1):154–162, 2000.

[Hu et coll., 2002] J. Hu, R. S. Kashi, D. Lopresti et G. T. Wilfong. Evaluating the performance of table processing algorithms. *International Journal on Document Analysis and Recognition*, 4(3):140–153, 2002.

[Hu et Ingold, 1993] T. Hu et R. Ingold. A mixed approach toward an efficient logical structure recognition from document images. *International Conference on Electronic Publishing: Document Manipulation and Typography*, 6(4):457–468, 1993.

[Hu et Xu, 2004] X. Hu et L. Xu. A comparative investigation on subspace dimension determination. *IEEE Transactions on Neural Networks*, 17(9):1051–1059, 2004.

[Hurst, 2001] M. Hurst. Layout and language: an efficient algorithm for detecting text blocks based on spatial and linguistic evidence. *SPIE - Document Recognition and Retrieval*, 4307(8):56–67, 2001.

[Hush et Horne, 1993] D. R. Hush et B. G. Horne. Progress in supervised neural networks: what's new since Lippmann? *IEEE Signal Processing Magazine*, 10(1):8–39, 1993.

[Hyvärinen, 1999] A. Hyvärinen. Survey on independent component analysis. *Neural Computing Surveys*, 2(1):94–128, 1999.

[Ingold, 1989] R. Ingold. Structures de documents et lecture optique: une nouvelle approche. *Thèse de l'École polytechnique Fédérale de Lausanne*, 1989.

[Ingold, 2002] R. Ingold. *Analyse et reconnaissance d'images de documents.* Techniques de l'Ingénieur, Référence H7020, 2002.

[Ingold et Armangil, 1991] R. Ingold et D. Armangil. A top-down document analysis method for logical structure recognition. *International Conference on Document Analysis and Recognition*, 1(1):41–49, 1991.

[Ishitani, 1999] Y. Ishitani. Logical structure analysis of document images based on emergent computation. *International Conference on Document Analysis and Recognition*, 1(5):189–192, 1999.

[Ishitani, 2003] Y. Ishitani. Document transformation system from papers to XML data based on pivot XML document method. *International Conference on Document Analysis and Recognition*, 1(7):250–255, 2003.

[Jain et Yu, 1998] A. K. Jain et B. Yu. Document representation and its application to page decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):294–308, 1998.

[Jain et Zhong, 1996] A. K. Jain et Y. Zhong. Page segmentation using texture analysis. *Pattern Recognition*, 29(5):743–770, 1996.

[Jolliffe, 2002] I. T. Jolliffe. *Principal Component Analysis, 2^{nd} edition.* Springer Verlag, 2002.

[Kacem et coll., 1999] A. Kacem, A. Belaïd et M. Ben Ahmed. EXTRAFOR: automatic extraction of mathematical formulas. *International Conference on Document Analysis and Recognition*, 1(5):527–530, 1999.

[Kanai et coll., 1995] J. Kanai, S. V. Rice, T. A. Nartker et G. Nagy. Automated evaluation of OCR zoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1):86–90, 1995.

[Kandel et coll., 2000] E. R. Kandel, J. H. Schwartz et T. M. Jessell. *Principles of Neural Science.* McGraw-Hill Medical, 2000.

[Kasturi et coll., 2002] R. Kasturi, L. O'Gorman et V. Govindaraju. Document image analysis: a primer. *Sadhana*, 27(1):3–22, 2002.

[Katz, 1987] S. M. Katz. Estimation of probabilities from spare data for the language model component of a speech recognizer. *Transaction on Acoustics, Speech and Signal Processing*, 3(35):400–401, 1987.

[Kim et Kim, 2000] G. Kim et S. Kim. Feature subset selection using genetic algorithms for handwritten digit recognition. *International Workshop on Frontiers in Handwriting Recognition*, 1(7):103–112, 2000.

[Kim et coll., 2001] J. Kim, D. X. Le et G. R. Thoma. Automated labeling in document images. *SPIE - Document Recognition and Retrieval*, 4307(8):111–122, 2001.

[Kim et coll., 2000] J. H. Kim, K. K. Kim et C. Y. Suen. An HMM-MLP hybrid model for cursive script recognition. *Pattern Analysis and Applications*, 3(4):314–324, 2000.

[Kim et Kim, 1998] Y.-S. Kim et W.-Y. Kim. Content-based trademark retrieval system using a visually salient feature. *Image and Vision Computing*, 16(12):931–939, 1998.

[Kise et coll., 1998] K. Kise, A. Sato et M. Iwata. Segmentation of page images using the area Voronoi diagram. *Computer Vision and Image Understanding*, 70(3):370–382, 1998.

[Koch et coll., 2005] G. Koch, L. Heutte et T. Paquet. Automatic extraction of numerical sequences in handwritten incoming mail documents. *Pattern Recognition Letters*, 26(8):1118–1127, 2005.

[Kohavi et John, 1997] R. Kohavi et G. John. Wrappers for feature selection. *Artificial Intelligence*, 97(1-2):272–324, 1997.

[Kohonen, 2001] T. Kohonen. *Self-Organizing Maps, Third Extended Edition.* Springer Series in Information Sciences, 2001.

[Kopec et Chou, 1994] G. E. Kopec et P. A. Chou. Document image decoding using Markov source models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):602–617, 1994.

[Kosmala et coll., 1999] A. Kosmala, G. Rigoll, S. Lavirotte et L. Pottier. On-line handwritten formula recognition using hidden Markov models and context dependent graph grammars. *International Conference on Document Analysis and Recognition*, 1(5):107–110, 1999.

[Kreich et coll., 1991] J. Kreich, A. Luhn et G. Maderlechner. An experimental environment for model based document analysis. *International Conference on Document Analysis and Recognition*, 1(1):50–58, 1991.

[Krishnamoorthy et coll., 1993] M. Krishnamoorthy, G. Nagy, S. Seth et M. Viswanathan. Syntactic segmentation and labeling of digitized pages from technical journals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(7):737–747, 1993.

[Küchler et Goller, 1996] A. Küchler et C. Goller. Inductive learning in symbolic domains using structure-driven recurrent neural networks. *Annual German Conference on Artificial Intelligence: Advances in Artificial Intelligence*, 1137(20):183–197, 1996.

[Lang et coll., 1990] K. J. Lang, A. H. Waibel et G. E. Hinton. A time-delay neural-network architecture for isolated word recognition. *IEEE Transactions on Neural Networks*, 3(1):23–44, 1990.

[Lapedes et Farber, 1987] A. Lapedes et R. Farber. Nonlinear signal processing using neural networks: prediction and signal modeling. *Technical Report LA-UR-87-2662, Los Alamos National Laboratories*, 1987.

[Le et coll., 1995a] D. X. Le, G. R. Thoma et H. Wechsler. Classification of binary document images into textual or nontextual data blocks using neural network models. *Machine Vision and Applications*, 8(5):289–504, 1995a.

[Le et coll., 1995b] D. X. Le, G. R. Thoma et H. Wechsler. Document image analysis using integrated image and neural processing. *International Conference on Document Analysis and Recognition*, 1(3):327–330, 1995b.

[Le Bourgeois et coll., 2001] F. Le Bourgeois, S. Souafi-Bensafi, J. Duong, M. Parizeau, M. Côté et H. Emptoz. Using statistical models in document images understanding. *International Workshop on Document Layout Interpretation and its Applications*, 2001.

[Le Cun, 1985] Y. Le Cun. Une procédure d'apprentissage pour réseau à seuil asymmetrique (A learning scheme for asymmetric threshold networks). *Proceedings of Cognitiva*, pages 599–604, 1985.

[Le Cun et coll., 1990a] Y. Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard et L. D. Jackel. Handwritten digit recognition with a back-propagation network. *Advances in Neural Information Processing Systems*, 2:396–404, 1990a.

[Le Cun et coll., 1998] Y. Le Cun, L. Bottou, Y. Bengio et P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Le Cun et coll., 1990b] Y. Le Cun, J. S. Denker, S. A. Solla, R. E. Howard et L. D. Jackel. Optimal brain damage. *Advances in Neural Information Processing Systems*, 2:598–605, 1990b.

[Lee et Srihari, 1995] D.-S. Lee et S. N. Srihari. Dynamic classifier combination using neural network. *SPIE - Document Recognition and Retrieval*, 2422(2):26–37, 1995.

[Liang et Doermann, 2002] J. Liang et D. S. Doermann. Logical labeling of document images using layout graph matching with adaptive learning. *International Workshop on Document Analysis Systems*, 2423(5):224–235, 2002.

[Liebowitz Taylor et coll., 1992] S. Liebowitz Taylor, R. Fritzson et J. A. Pastor. Extraction of data from preprinted forms. *Machine Vision and Applications*, 5(3):211–222, 1992.

[Lin et coll., 1997] C. C. Lin, Y. Niwa et S. Narita. Logical structure analysis of book document images using contents information. *International Conference on Document Analysis and Recognition*, 2(4):1048–1054, 1997.

[Liu et coll., 2003] C.-L. Liu, K. Nakashima, H. Sako et H. Fujisawa. Handwritten digit recognition: benchmarking of state-of-the-art techniques. *Pattern Recognition*, 36(10):2271–2285, 2003.

[Liu et coll., 2002] G.-S. Liu, Y.-C. Wang et P.-H. Hu. Analyzing document logic structure by machine learning. *International Conference on Machine Learning and Cybernetics*, 1(1):179–183, 2002.

[Lladós et coll., 2001a] J. Lladós, E. Martí et J. J. Villanueva. Symbol recognition by error-tolerant subgraph matching between region adjacency graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10):1137–1143, 2001a.

[Lladós et coll., 2001b] J. Lladós, E. Valveny, G. Sánchez et E. Martí. Symbol recognition: current advances and perspectives. *International Workshop on Graphics Recognition, Algorithms and Applications*, 2390(4):104–127, 2001b.

[Logar et coll., 1993] A. M. Logar, E. M. Corwin et W. J. B. Oldham. A comparison of recurrent neural network learning algorithms. *IEEE Transactions on Neural Networks*, 2:1129–1134, 1993.

[Lu et coll., 1998] Z. Lu, Z. Chi et W.-C. Siu. Length estimation of digit strings using neural networks with structure based features. *SPIE - Journal of Electronic Imaging*, 7(1):79–85, 1998.

[MacQueen, 1967] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. *Berkeley Symposium on Mathematical Statistics and Probability*, 1(5):281–297, 1967.

[Malerba et coll., 2007] D. Malerba, M. Berardi et M. Ceci. *Machine Learning for Reading Order Detection in Document Image Understanding.* Machine learning in Document Analysis and Recognition, Springer Verlag, 2007.

[Mao et coll., 2003] S. Mao, A. Rosenfeld et T. Kanungo. Document structure analysis algorithms: a literature survey. *SPIE - Document Recognition and Retrieval*, 5010(10):197–207, 2003.

[Marinai et coll., 2005] S. Marinai, M. Gori et G. Soda. Artificial neural networks for document analysis and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(1):23–35, 2005.

[Marques et coll., 2005] F. D. Marques, L. d. F. Rodrigues de Souza, D. C. Rebolho, A. S. Caporali et E. M. Belo. Application of time-delay neural and recurrent neural networks for the identification of a hingeless helicopter blade flapping and torsion motions. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 27(2):97–103, 2005.

[Martin, 1993] G. L. Martin. Centered-object integrated segmentation and recognition of overlapping handprinted characters. *Neural Computation*, 5(3):419–429, 1993.

[Martin et Pittman, 1989] G. L. Martin et J. A. Pittman. Recognizing hand-printed letters and digits. *Advances in Neural Information Processing Systems*, 2:405–414, 1989.

[Martin et Bellissant, 1991] P. Martin et C. Bellissant. Low-level analysis of music drawing images. *International Conference on Document Analysis and Recognition*, 1(1):417–425, 1991.

[McClelland et Rumelhart, 1981] J. L. J. McClelland et D. E. Rumelhart. An interactive activation model of context effects in letter perception. *Psychological Review*, 88(5):375–407, 1981.

[McCulloch et Pitts, 1943] W. S. McCulloch et W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.

[Minsky et Papert, 1969] M. L. Minsky et S. A. Papert. *Perceptrons.* Cambridge, MIT Press, 1969.

[Mirowski et coll., 2007] P. W. Mirowski, D. Madhavan et Y. Le Cun. Time-delay neural networks and independent component analysis for eeg-based prediction of epileptic seizures propagation. *AAAI/SIGART Doctoral Consortium*, pages 1892–1983, 2007.

[Moody et Darken, 1989] J. Moody et C. Darken. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1(1):289–303, 1989.

[Morton, 1969] J. Morton. Interaction of information in word recognition. *Psychological Review*, 76(2):165–178, 1969.

[Mui et coll., 1994] L. Mui, A. Agarwal, A. Gupta et P. S.-P. Wang. An adaptive modular neural network with application to unconstrained character recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 8(5):1189–1204, 1994.

[Mulgaonkar, 1986] P. G. Mulgaonkar. Automatic detection of address blocks on irregular mail pieces. *Conference on Computer Vision and Pattern Recognition*, pages 672–674, 1986.

[Nadler, 1984] M. Nadler. A survey of document segmentation and coding techniques. *Computer Vision, Graphics, and Image Processing*, 28:240–262, 1984.

[Nagy, 1992] G. Nagy. Teaching a computer to read. *International Conference on Pattern Recognition*, 2(11):225–229, 1992.

[Nagy, 2000] G. Nagy. Twenty years of document image analysis in PAMI. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):38–62, 2000.

[Nagy et Seth, 1984] G. Nagy et S. Seth. Hierarchical representation of optically scanned documents. *International Conference on Pattern Recognition*, 1(7):347–349, 1984.

[Nagy et coll., 1992] G. Nagy, S. Seth et M. Viswanathan. A prototype document image analysis system for technical journals. *Computer*, 25(7):10–22, 1992.

[Narendra et Parthasarathy, 1990] K. S. Narendra et K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1(1):4–27, 1990.

[Nicolas et coll., 2004] S. Nicolas, T. Paquet et L. Heutte. Enriching historical manuscripts: the Bovary project. *International Workshop on Document Analysis Systems*, 3163(6):135–146, 2004.

[Nielson et Barrett, 2003] H. E. Nielson et W. A. Barrett. Consensus-based table form recognition. *International Conference on Document Analysis and Recognition*, 1(7):906–910, 2003.

[Niyogi et Srihari, 1995] D. Niyogi et S. N. Srihari. Knowledge-based derivation of document logical structure. *International Conference on Document Analysis and Recognition*, 1(3):472–475, 1995.

[O'Gorman, 1993] L. O'Gorman. The document spectrum for page layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1162–1173, 1993.

[O'Gorman et Kasturi, 1996] L. O'Gorman et R. Kasturi. *Document Image Analysis*. Institute of Electrical and Electronics Engineers, 1996.

[Okamoto et Takahashi, 1993] M. Okamoto et M. Takahashi. A hybrid page segmentation method. *International Conference on Document Analysis and Recognition*, 1(2):743–746, 1993.

[Palaniappan et coll., 2000] R. Palaniappan, P. Raveendran et S. Omatu. New invariant moments for non-uniformly scaled images. *Pattern Analysis and Applications*, 3(2):78–87, 2000.

[Parmentier et Belaïd, 1997] F. Parmentier et A. Belaïd. Logical structure recognition of scientific bibliographic references. *International Conference on Document Analysis and Recognition*, 2(4):1072–1076, 1997.

[Pasquer et coll., 2000] L. Pasquer, É. Anquetil et G. Lorette. Système itératif d'interprétation multicontextuelle pour la lecture d'écriture manuscrite. *Reconnaissance des Formes et Intelligence Artificielle*, 1(12):347–356, 2000.

[Pavlidis et Zhou, 1992] T. Pavlidis et J. Zhou. Page segmentation and classification. *Graphical Models and Image Processing*, 54(6):484–496, 1992.

[Pearl, 1988] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: networks of Plausible Inference*. Morgan-Kaufmann Series in Representation and Reasoning, 1988.

[Pearlmutter, 1995] B. A. Pearlmutter. Gradient calculations for dynamic recurrent neural networks: a survey. *IEEE Transactions on Neural Networks*, 6(5):1212–1228, 1995.

[Pfister et coll., 2000] M. Pfister, S. Behnke et R. Rojas. Recognition of handwritten ZIP codes in a real-world non-standard-letter sorting system. *Applied Intelligence*, 12(1-2):95–115, 2000.

[Pineda, 1987] F. J. Pineda. Generalization of back-propagation to recurrent neural networks. *Physical Review Letters*, 19(59):2229–2232, 1987.

[Pineda, 1988] F. J. Pineda. Dynamics and architecture for neural computation. *Journal of Complexity*, 4(3):216–245, 1988.

[Pineda, 1989] F. J. Pineda. Recurrent back-propagation and the dynamical approach to adaptive neural computation. *Neural Computation*, 1(1):161–172, 1989.

[Piwowarski et coll., 2002] B. Piwowarski, L. Denoyer et P. Gallinari. Un modèle pour la recherche d'information sur des documents structurés. *Journées Internationales d'Analyse Statistique des Données Textuelles*, 1(6), 2002.

[Quinlan, 1993] J. R. Quinlan. *C4.5: programs for Machine Learning*. Morgan Kaufmann Publishers Inc, 1993.

[Quint et Vatton, 1986] V. Quint et I. Vatton. Grif: an interactive system for structured document manipulation. *International Conference on Text Processing and Document Manipulation*, 1(1):200–213, 1986.

[Rabiner, 1989] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[Rahman et Fairhurst, 1999] A. F. R. Rahman et M. C. Fairhurst. Enhancing multiple expert decision combination strategies through exploitation of a priori information sources. *IEEE Proceedings - Vision, Image, and Signal Processing*, 146(1):40–49, 1999.

[Ramel et coll., 2003] J.-Y. Ramel, M. Crucianu, N. Vincent et C. Faure. Detection, extraction and representation of tables. *International Conference on Document Analysis and Recognition*, 1(7):374–378, 2003.

[Rangoni et Belaïd, 2005] Y. Rangoni et A. Belaïd. Data categorization for a context return applied to logical document structure recognition. *International Conference on Document Analysis and Recognition*, 1(8):297–301, 2005.

[Rangoni et Belaïd, 2006] Y. Rangoni et A. Belaïd. Document logical structure analysis based on perceptive cycles. *International Workshop on Document Analysis Systems*, 3872(7):117–128, 2006.

[Rangoni et Belaïd, 2006] Y. Rangoni et A. Belaïd. Reconnaissance de structures logiques par un réseau de neurones transparent. *Colloque International Francophone sur l'Écrit et le Document*, 1(9):1–6, 2006.

[Reicher, 1969] G. M. Reicher. Perceptual recognition as a function of meaningfulness of stimulus material. *Journal of Experimental Psychology*, 81(2):275–280, 1969.

[Rice et coll., 1999] S. V. Rice, G. Nagy et T. A. Nartker. *Optical Character Recognition: an Illustrated Guide to the Frontier*. Kluwer Academic Publishers, 1999.

[Rivals et Personaz, 2003] I. Rivals et L. Personaz. MLPs (mono-layer polynomials and multi-layer perceptrons) for nonlinear modeling. *Journal of Machine Learning Research*, 3:1383–1398, 2003.

[Rohwer et Forrest, 1987] R. Rohwer et B. Forrest. Training time-dependence in neural networks. *International Conference on Neural Networks*, 2(1):701–708, 1987.

[Rondel et Bure, 1995] N. Rondel et G. Bure. Cooperation of multilayer Perceptrons for the estimation of skew angle in text documents images. *International Conference on Document Analysis and Recognition*, 2(3):1141–1144, 1995.

[Rosenblatt, 1958] F. Rosenblatt. The Perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 6(65):386–408, 1958.

[Rosenfeld et coll., 1976] A. Rosenfeld, R. A. Hummel et S. W. Zucker. Scene labeling by relaxation operations. *IEEE Transactions on Systems, Man and Cybernetics*, 6(6):420–433, 1976.

[Rumelhart et coll., 1986] D. E. Rumelhart, G. E. Hinton et R. J. Williams. Learning internal representations by error propagation. *Parallel Data Processing: Explorations in the Microstructure of Cognition*, 1:318–362, 1986.

[Sainz Palmero et coll., 1996] G. I. Sainz Palmero, J. M. Cano Izquierdo, Y. A. Dimitriadis et J. López Coronado. A new neuro-fuzzy system for logical labeling of documents. *International Conference on Pattern Recognition*, 4(18):431–435, 1996.

[Sainz Palmero et Dimitriadis, 1999] G. I. Sainz Palmero et Y. A. Dimitriadis. Structured document labeling and rule extraction using new recurrent fuzzy-neural systems. *International Conference on Document Analysis and Recognition*, 1(5):181–184, 1999.

[Saitoh et coll., 1993] T. Saitoh, M. Tachikawa et T. Yamaai. Document image segmentation and text area ordering. *International Conference on Document Analysis and Recognition*, 1(2):323–329, 1993.

[Sankar et Mammone, 1992] A. Sankar et R. J. Mammone. *Neural tree networks*. Neural networks: theory and applications, Academic Press Professional, 1992.

[Saon et Belaïd, 1997] G. Saon et A. Belaïd. High performance unconstrained word recognition system combining HMMs and Markov random fields. *International Journal of Pattern Recognition and Artificial Intelligence*, 11(5):771–788, 1997.

[Sarkar et Nagy, 2005] P. Sarkar et G. Nagy. Style consistent classification of isogenous patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(1):88–98, 2005.

[Schapire, 1990] R. E. Schapire. Strength of weak learnability. *Journal of Machine Learning*, 5(2):197–227, 1990.

[Schenkel et coll., 1994] M. Schenkel, I. Guyon et D. Henderson. On-line cursive script recognition using time delay neural networks and hidden Markov models. *International Conference on Acoustics, Speech, and Signal Processing*, 2:637–640, 1994.

[Segui, 1991] J. Segui. *La Reconnaissance Visuelle de Mots. La Reconnaissance des Mots dans Différentes Modalités Sensoriellees: étude de Psycholinguistique Cognitive*. Presses Universitaires de France, 1991.

[Sejnowski et Rosenberg, 1987] T. J. Sejnowski et C. R. Rosenberg. Parallel networks that learn to pronounce English text. *Complex Systems*, 1(1):145–168, 1987.

[Simard et Le Cun, 1992] P. Simard et Y. Le Cun. Reverse tdnn: an architecture for trajectory generation. *Advances in Neural Information Processing Systems*, 4:579–588, 1992.

[Singh et coll., 2000] R. Singh, V. Cherkassky et N. P. Papanikolopoulos. Self-organizing maps for the skeletonization of sparse shapes. *IEEE Transactions on Neural Networks*, 11(1):241–248, 2000.

[Snoussi Maddouri, 2003] S. Snoussi Maddouri. Modèle perceptif neuronal à vision globale-locale pour la reconnaissance de mots arabes omni-scripteurs. *Thèse de l'École Nationale d'Ingénieurs de Tunis*, 2003.

[Snoussi Maddouri et coll., 2000] S. Snoussi Maddouri, H. Amiri et A. Belaïd. Local normalization towards global recognition of Arabic handwritten script. *International Workshop on Document Analysis Systems*, 1(4):1–13, 2000.

[Snoussi Maddouri et coll., 2002] S. Snoussi Maddouri, H. Amiri, A. Belaïd et C. Choisy. Combination of local and global vision modelling for Arabic handwritten words recognition. *International Workshop on Frontiers in Handwriting Recognition*, 1(8):128–135, 2002.

[Souafi-Bensafi et coll., 2002] S. Souafi-Bensafi, M. Parizeau, F. Le Bourgeois et H. Emptoz. Bayesian networks classifiers applied to documents. *International Conference on Pattern Recognition*, 1(16):483–486, 2002.

[Sperduti et Starita, 1997] A. Sperduti et A. Starita. Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks*, 8(3):714–735, 1997.

[Sperduti et coll., 1995] A. Sperduti, A. Starita et C. Goller. Learning distributed representations for the classification of terms. *Proceedings of International Joint Conference on Artificial Intelligence*, 1(40):509–515, 1995.

[Spitz, 1991] A. L. Spitz. Style-directed document recognition. *International Conference on Document Analysis and Recognition*, 2(1):611–619, 1991.

[Srihari et coll., 1999] S. N. Srihari, W.-J. Yang et V. Govindaraju. Information theoretic analysis of postal address fields for automatic address interpretation. *International Conference on Document Analysis and Recognition*, 1(5):309–312, 1999.

[Srihari et Zack, 1986] S. N. Srihari et G. W. Zack. Document image analysis. *International Conference on Pattern Recognition*, 1(8):434–436, 1986.

[Steinherz et coll., 1999] T. Steinherz, E. Rivlin et N. Intrator. Offline cursive script word recognition, a survey. *International Journal on Document Analysis and Recognition*, 2(2):90–110, 1999.

[Stoppiglia et coll., 2003] H. Stoppiglia, G. Dreyfus, R. Dubois et Y. Oussar. Ranking a random feature for variable and feature selection. *Journal of Machine Learning Research*, 3:1399–1414, 2003.

[Strathy et Suen, 1995] N. W. Strathy et C. Y. Suen. A new system for reading handwritten ZIP codes. *International Conference on Document Analysis and Recognition*, 1(3):74–77, 1995.

[Strouthopoulos et Papamarkos, 1998] C. Strouthopoulos et N. Papamarkos. Text identification for document image analysis using a neural network. *Image and Vision Computing*, 16(12):879–896, 1998.

[Stubberud et coll., 1995] P. Stubberud, J. Kanai et V. Kalluri. Adaptive image restoration of text images that contain touching or broken characters. *International Conference on Document Analysis and Recognition*, 1(3):778–781, 1995.

[Sugiyama et coll., 1991] M. Sugiyama, H. Sawai et A. H. Waibel. Review of TDNN (Time Delay Neural Network) architectures for speech recognition. *IEEE International Symposium on Circuits and Systems*, 1:582–585, 1991.

[Summers, 1995a] K. Summers. Near-wordless document structure classification. *International Conference on Document Analysis and Recognition*, 1(3):462–465, 1995a.

[Summers, 1995b] K. Summers. Toward a taxonomy of logical document structures. *Proceedings of the Dartmouth Institute for Advanced Graduate Studies*, 2(4):124–133, 1995b.

[Sun et coll., 2004] Z. Sun, G. Bebis et R. Miller. Object detection using feature subset selection. *Pattern Recognition*, 37(11):2165–2176, 2004.

[Szilas, 1995] N. Szilas. Apprentissage dans les réseaux mécaniques et étude de leurs interactions avec l'environnement. *Thèse de l'Institut National Polytechnique de Grenoble*, 1995.

[Taft, 1991] M. Taft. *Essays in Cognitive Psychology, Reading and the Mental Lexicon.* Psychology Press, 1991.

[Tang et coll., 1999] Y. Y. Tang, M. Cheriet, J. Liu, J. N. Said et C. Y. Suen. *Document Analysis and Recognition by Computers.* Handbook of Pattern Recognition and Computer Vision, 1999.

[Tang et coll., 1997] Y. Y. Tang, H. Ma, J. Liu, B. F. Li et D. Xi. Multiresolution analysis in extraction of reference lines from documents with gray level background. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(8):921–926, 1997.

[Tateisi et Itoh, 1994] Y. Tateisi et N. Itoh. Using stochastic syntactic analysis for extracting a logical structure from a document image. *International Conference on Pattern Recognition*, 2(12):391–394, 1994.

[Taylor et Taylor, 1983] I. Taylor et M. M. Taylor. *The Psychology of Reading*. New York: Academic Press, 1983.

[Tetko et coll., 1995] I. V. Tetko, D. J. Livingstone et A. I. Luik. Neural network studies. 1. comparison of overfitting and overtraining. *Journal of Chemical Information and Computer Sciences*, 35(5):826–833, 1995.

[Tokuyasu et Chou, 2001] T. A. Tokuyasu et P. A. Chou. Turbo recognition: a statistical approach to layout analysis. *SPIE - Document Recognition and Retrieval*, 4307(8):123–129, 2001.

[Torkkola, 2003] K. Torkkola. Feature extraction by non-parametric mutual information maximization. *Journal of Machine Learning Research*, 3:1415–1438, 2003.

[Toyozumi et coll., 2004] K. Toyozumi, N. Yamada, K. Mase, T. Kitasaka, K. Mori, Y. Suenaga et T. Takahashi. A study of symbol segmentation method for handwritten mathematical formula recognition using mathematical structure information. *International Conference on Pattern Recognition*, 2(17):630–633, 2004.

[Tsujimoto et Asada, 1990] S. Tsujimoto et H. Asada. Understanding multi-articled documents. *International Conference on Pattern Recognition*, 1(10):551–556, 1990.

[Tsuruoka et coll., 2001] S. Tsuruoka, C. Hirano, T. Yoshikawa et T. Shinogi. Image-based structure analysis for a table of contents and conversion to XML documents. *International Workshop on Document Layout Interpretation and its Applications*, 1(2):59–62, 2001.

[Turolla et coll., 1995] E. Turolla, Y. Belaïd et A. Belaïd. Line and cell searching in tables or forms. *International Conference on Image Analysis and Processing*, 974(8):509–514, 1995.

[Vajda et coll., 2006] S. Vajda, Y. Rangoni, H. Cecotti et A. Belaïd. A fast learning strategy using data selection for feedforward neural networks. *International Workshop on Frontiers in Handwriting Recognition*, 1(10):14–150, 2006.

[Vapnik, 1995] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.

[Vinciarelli, 2002] A. Vinciarelli. A survey on off-line cursive word recognition. *Pattern Recognition*, 35(7):1433–1446, 2002.

[Wahl et coll., 1982] F. M. Wahl, K. Y. Wong et R. G. Kasey. Block segmentation and text extraction in mixed text/image documents. *Computer Graphics and Image Processing*, 20(4):375–390, 1982.

[Walischewski, 1997] H. Walischewski. Automatic knowledge acquisition for spatial document interpretation. *International Conference on Document Analysis and Recognition*, 1(4):243–247, 1997.

[Wan, 1993] E. A. Wan. Finite impulse response neural networks with applications in time series prediction. *Phd Thesis, Standford University*, 1993.

[Wan, 1994] E. A. Wan. *Time Series Prediction by Using a Connectionist Network with Internal Delay Lines*, vol. 17. Time Series Prediction: forecasting the Future and Understanding the Past, Addison-Wesley, 1994.

[Wang et Srihari, 1989] D. Wang et S. N. Srihari. Classification of newspaper image blocks using texture analysis. *Computer Vision, Graphics, and Image Processing*, 47(3):327–352, 1989.

[Wang et Jean, 1993] J. Wang et J. Jean. Segmentation of merged characters by neural networks and shortest path. *Symposium on Applied Computing: States of the Art and Practice*, pages 762–769, 1993.

[Watanabe et coll., 1995] T. Watanabe, Q. Luo et N. Sugie. Layout recognition of multi-kinds of table-form documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(4):432–445, 1995.

[Weindorf, 2001] M. Weindorf. Structure based interpretation of unstructured vector maps. *International Workshop on Graphics Recognition, Algorithms and Applications*, 2390(4):190–199, 2001.

[Wenyin et Dori, 1999] L. Wenyin et D. Dori. From raster to vectors: extracting visual information from line drawings. *Pattern Analysis and Applications*, 2(1):10–21, 1999.

[Wenzel et Maus, 2001] C. Wenzel et H. Maus. Leveraging corporate context within knowledge-based document analysis and understanding. *International Journal on Document Analysis and Recognition*, 3(4):248–260, 2001.

[Werbos, 1994] P. J. Werbos. *The Roots of Backpropagation: from Ordered Derivatives to Neural Networks and Political Forecasting*. Wiley-Interscience, 1994.

[Weston et coll., 2003] J. Weston, A. Elisseff, B. Schölkopf et M. Tipping. Use of the zero norm with linear models and kernel methods. *Journal of Machine Learning Research*, 3(8):1439–1461, 2003.

[Whichello et Yan, 1996] A. P. Whichello et H. Yan. Linking broken character borders with variable sized masks to improve recognition. *Pattern Recognition*, 29(8):1429–1435, 1996.

[Wöhler et Anlauf, 1999] C. Wöhler et J. K. Anlauf. A time delay neural network algorithm for estimating image-pattern shape and motion. *Image and Vision Computing*, 17(3-4):281–294, 1999.

[Widrow et Hoff, 1960] B. Widrow et M. E. Hoff. Adaptive switching circuits. *Western Electronic Show and Convention*, 4:96–104, 1960.

[Williams et Zipser, 1989] R. J. Williams et D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(1):270–280, 1989.

[Wolf et coll., 1997] M. Wolf, H. Niemann et W. Schmidt. Fast address block location on handwritten and machine printed mail-piece images. *International Conference on Document Analysis and Recognition*, 2(4):753–757, 1997.

[Xu et coll., 2003] Q. Xu, L. Lam et C. Y. Suen. Automatic segmentation and recognition system for handwritten dates on Canadian bank cheques. *International Conference on Document Analysis and Recognition*, 1(7):704–708, 2003.

[Yamashita et coll., 1991] A. Yamashita, T. Amasno, H. Takahashi et K. Toyokawa. A model based layout understanding method for the document recognition system. *International Conference on Document Analysis and Recognition*, 1(1):130–138, 1991.

[Yanadume et coll., 2004] S. Yanadume, Y. Mekada, I. Ide et H. Murase. Recognition of very low-resolution characters from motion images captured by a portable digital camera. *Advances in Multimedia Information Processing*, 3331(5):247–254, 2004.

[Yanikoglu et Vincent, 1998] B. A. Yanikoglu et L. Vincent. Pink Panther: a complete environment for ground-truthing and benchmarking document page segmentation. *Pattern Recognition*, 31(9):1191–1204, 1998.

[You et Kim, 2003] D. You et G. Kim. An approach for locating segmentation points of handwritten digit strings using a neural network. *International Conference on Document Analysis and Recognition*, 1(7):142–146, 2003.

[Yu et Liu, 2003] L. Yu et H. Liu. Feature selection for high-dimensional data: a fast correlation-based filter solution. *International Conference on Machine Leaning*, 1(12):856–863, 2003.

[Zanibbi et coll., 2002] R. Zanibbi, D. Blostein et J. R. Cordy. Recognizing mathematical expressions using tree transformation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11):1455–1467, 2002.

[Zhang et coll., 2007] X.-W. Zhang, M. R. Lyu et G.-Z. Dai. Extraction and segmentation of tables from Chinese ink documents based on a matrix model. *Pattern Recognition*, 40(7):1855–1867, 2007.

[Zhong et coll., 2006] M. Zhong, S. Sharma et P. Lingras. Genetically-designed time delay neural networks for multiple-interval urban freeway traffic flow forecasting. *Neural Information Processing - Letters and Reviews*, 10(8-9):201–209, 2006.

[Zhou et coll., 2001] J. Zhou, C. Y. Suen et K. Liu. A feedback-based approach for segmenting handwritten legal amounts on bank cheques. *International Conference on Document Analysis and Recognition*, 1(6):887–891, 2001.

Index des auteurs

Agarwal, A. 50
Ahmed, P. 49
Aiello, M. 20, 22
Akindele, O. T. 18, 20, 26, 27, 50
Al-Sadoun, H. 50
Alam, H. 1
Almeida, L. B. 105
Altamura, O. 19, 20
Alusse, A. 1
Amano, A. 6
Amasno, T. 13, 16, 26, 27
Amin, A. 50
Amiri, H. 2, 42, 43
André, J. 8
Anigbogu, J. C. 9, 50
Anlauf, J. K. 111
Anquetil, É. 33
Armangil, D. 16, 18, 26, 27
Arthur, D. 97
Asada, H. 13, 16, 26, 27
Asada, N. 6
Atalay, V. 82
Azzabou, N. 21, 22

Babaguchi, N. 22, 24
Baccino, T. 32
Badran, F. 43
Baird, H. S. 23, 24
Balci, K. 82
Baldi, P. 104
Bapst, F. 18
Barrett, W. A. 23, 24
Basile, T. M. A. 19, 20
Bebis, G. 83, 100
Becker, C. A. 35
Behnke, S. 111
Belaïd, A. 1, 2, 6, 9, 15, 16, 18, 20–24, 26, 27, 42, 43, 49, 50, 55, 64, 67, 70, 93
Belaïd, Y. 1, 6, 21, 22
Bellissant, C. 49
Belo, E. M. 111
Ben Ahmed, M. 6
Benahmed, N. 79
Benet, N. 6
Bengio, Y. 50, 140, 141
Bennett, K. P. 79
Berardi, M. 10, 19–22
Besagni, D. 6
Bi, J. 79
Bleisinger, R. 14, 16
Bloechle, J.-L. 1
Blostein, D. 6, 17
Blum, A. 76
Bortolozzi, F. 79
Boser, B. 34
Bottou, L. 50, 140, 141
Bouriel, K. 43
Bramall, P. E. 33
Breiman, L. 66
Breneman, C. M. 79
Breuel, T. M. 50
Bromley, J. 111
Brown, B. 8
Brugger, R. 18, 20, 26, 27
Buchman, M. 8
Bure, G. 49
Burge, M. 6
Burkowski, F. J. 1

Cano Izquierdo, J. M. 20
Caporali, A. S. 111
Carreira-Perpiñán, M. Á. 80
Casasent, D. 7
Casey, R. G. 34, 50
Cattell, R. B. 92
Cattoni, R. 9
Ceci, M. 10, 19–22
Cecotti, H. 1, 50, 64
Ceheux, G. R. 1
Cesarini, F. 6, 49
Chang, M.-T. 6
Chaudhuri, A. R. 6
Chaudhuri, B. B. 6, 49
Chellappa, R. 23, 24, 49, 82
Chen, S.-Y. 6
Chenevoy, Y. 9, 15, 16, 24, 26, 27
Cheriet, M. 2, 10, 31
Cherkassky, V. 49

Chi, Z. 49, 50
Choisy, C. 2, 43, 50
Chou, P. A. 49
Clarke, C. L. A. 1
Cohen, E. 23, 24
Coianiz, T. 9
Colé, P. 32
Coltheart, M. 35
Conway, A. 16–18, 26, 27
Cordy, J. R. 6
Cormack, G. V. 1
Cornuéjols, A. 43
Corwin, E. M. 106
Côté, M. 2, 20, 22, 29, 31, 40, 51, 53, 71, 74, 113, 123
Coüasnon, B. 17, 18
Crucianu, M. 6

Dai, G.-Z. 6
Darken, C. 111
Datta, A. 49
de Oliveira, L. E. S. 79
Delalandre, M. 6
Dengel, A. R. 12–14, 16, 19, 20, 26, 27
Denker, J. S. 34, 74, 79
Denoyer, L. 1
Derrien-Péden, D. 13, 16, 26, 27
Di Mauro, N. 19, 20
Diligenti, M. 50
Dimitriadis, Y. A. 20, 22
Doermann, D. S. 6–8, 19, 20, 23, 24, 49
Dori, D. 6, 8
Dreyfus, G. 43, 79, 80
Dubiel, F. 19, 20, 26, 27
Dubois, R. 79, 80
Duong, J. 20, 22

Eastwood, B. 50
Edwards, J. J. 19
Elisseff, A. 79
Embrechts, M. 79
Emptoz, H. 20, 22
Esposito, F. 19, 20, 22, 24
Etemad, K. 23, 24, 49, 82

Fahlman, S. E. 120
Fahmy, H. 17
Fairhurst, M. C. 50
Falk, I. 55, 70
Farber, R. 111
Faure, C. 6
Fein, F. 14, 16
Ferilli, S. 19, 20
Fischer, S. 50
Fisher, J. L. 13, 16, 26, 27
Fletcher, L. A. 23, 24
Forrest, B. 105
Fortune, S. J. 23, 24
Francesconi, E. 6
Frasconi, P. 6, 125
Freund, Y. 66
Fritzson, R. 49
Frydrych, M. 7
Fujisawa, H. 50
Furuta, R. 8

Gallinari, P. 1
Garain, U. 6
Ghosh, R. P. 6
Goller, C. 119, 120, 125
Gonzalez, R. C. 19
Gordon, M. 43
Gori, M. 6, 28, 49, 50, 125
Govindaraju, V. 23, 24, 26, 27, 53
Grbavec, A. 17
Gupta, A. 50
Guyon, I. 111
Gyohten, K. 22, 24

Hadjamar, H. 1
Hadjar, K. 1, 19, 20
Haffner, P. 50, 140, 141
Hall, M. A. 80
Hamrouni, K. 43
Hamza, H. 49
Hancock, E. R. 11
Haralick, R. M. 8, 10
Hartono, R. 1
Harvey, A. 50
Hassibi, B. 74
Henderson, D. 34, 111
Hertz, J. 105
Heutte, L. 6, 23
Higgins, C. A. 33
Hinton, G. E. 45, 107, 111
Hirano, C. 6
Hitz, O. 19, 20
Hoch, R. 14, 16
Hoff, M. E. 44
Hones, F. 14, 16
Hopfield, J. J. 106
Horne, B. G. 111
Howard, R. E. 34, 74, 79
Hérault, L. 43
Héroux, P. 19, 20
Hu, J. 6
Hu, P.-H. 19, 20
Hu, T. 14, 16
Hu, X. 91
Hubbard, W. 34
Hull, J. J. 23, 24
Hummel, R. A. 20

Hurst, M. 18, 20
Hush, D. R. 111
Hyvärinen, A. 80

Ide, I. 82
Ingold, R. 1, 8, 14, 16, 18–20, 26, 27
Intrator, N. 34
Ishitani, Y. 11, 13, 14, 16, 26, 27, 71
Itoh, N. 17, 18, 26, 27
Iwata, M. 23, 24

Jackel, L. D. 34, 74, 79
Jain, A. K. 23, 24, 49
Jean, J. 50
Jennings, A. 50
Jessell, T. M. 143
John, G. 79
Jolliffe, I. T. 81, 88
Jones, S. E. 23, 24

Kacem, A. 6
Kakusho, K. 22, 24
Kalluri, V. 49
Kanai, J. 49, 63
Kandel, E. R. 143
Kanungo, T. 24, 25
Kasey, R. G. 23, 24
Kashi, R. S. 6
Kasturi, R. 23, 24, 53
Katz, S. M. 56
Kim, G. 50, 79
Kim, J. 11–13, 26, 27, 71
Kim, J. H. 50
Kim, K. K. 50
Kim, S. 79
Kim, W.-Y. 6
Kim, Y.-S. 6
Kise, K. 23, 24
Kitahashi, T. 22, 24
Kitasaka, T. 6
Klein, B. 12, 13
Koch, G. 23
Kohavi, R. 79
Kohonen, T. 49
Kopec, G. E. 49
Kosmala, A. 6
Kreich, J. 12, 13, 26, 27
Krishnamoorthy, M. 16, 18, 26, 27, 117
Krogh, A. 105
Küchler, A. 119, 120, 125
Kumar, A. 1

Lalanne, D. 1
Lam, L. 50
Lang, K. J. 107, 111
Langley, P. 76
Lapedes, A. 111
Lastri, M. 49
Lavirotte, S. 6
Le Bourgeois, F. 20
Le Cun, Y. 34, 45, 50, 74, 79, 111, 140, 141
Le, D. X. 11–13, 21, 22, 26, 27, 71
Lebiere, C. 120
Lecolinet, É. 2, 31, 34, 50
Lecoutier, Y. 19, 20
Lee, D.-S. 50
Li, B. F. 22, 24
Liang, J. 19, 20
Liebowitz Taylor, S. 49
Likforman-Sulem, L. 21, 22
Lin, C. C. 11, 13, 26, 27
Lingras, P. 111
Liu, C.-L. 50
Liu, G.-S. 19, 20
Liu, H. 80
Liu, J. 10, 22, 24
Liu, K. 50
Livingstone, D. J. 48
Lladós, J. 6, 17
Logar, A. M. 106
López Coronado, J. 20
Lopresti, D. 6
Lorette, G. 33
Lu, Z. 50
Luhn, A. 12, 13, 26, 27
Luik, A. I. 48
Luo, Q. 6
Lyu, M. R. 6

Ma, H. 22, 24
MacQueen, J. B. 97
Maderlechner, G. 12, 13, 26, 27
Madhavan, D. 111
Maggini, M. 50
Malerba, D. 10, 19–22, 24
Mammone, R. J. 120
Mao, S. 24, 25
Marinai, S. 6, 28, 49, 50
Marques, F. D. 111
Martí, E. 6, 17
Martin, G. L. 34, 111
Martin, P. 49
Martinelli, E. 50
Martinez, J. 43
Mase, K. 6
Maus, H. 15, 16
McClelland, J. L. J. 2, 31, 37, 38, 40, 50, 53, 71, 118, 122, 123
McCulloch, W. S. 143
Mekada, Y. 82
Messelodi, S. 9
Miclet, L. 43
Miller, R. 83, 100

Minsky, M. L. 44
Mirowski, P. W. 111
Modena, C. M. 9
Monagan, G. 6
Monz, C. 20, 22
Moody, J. 111
Mori, K. 6
Morita, M. 79
Morton, J. 33
Mui, L. 50
Mukunoki, M. 6
Mulgaonkar, P. G. 23
Murase, H. 82

Nadler, M. 9
Nagy, G. 5–7, 14, 16–18, 23, 24, 26, 27, 50, 56, 63, 117, 119
Nakashima, K. 50
Narendra, K. S. 106
Narita, S. 11, 13, 26, 27
Nartker, T. A. 56, 63
Nicolas, S. 6
Nielson, H. E. 23, 24
Niemann, H. 22, 24
Niwa, Y. 11, 13, 26, 27
Niyogi, D. 12, 13, 16, 26, 27

Ogier, J.-M. 6
O'Gorman, L. 23, 24, 53
Okamoto, M. 23, 24
Oldham, W. J. B. 106
Omatu, S. 49
Oussar, Y. 79, 80

Palaniappan, R. 49
Palmer, R. G. 105
Panchèvre, J.-L. 6, 21, 22
Papamarkos, N. 49
Papanikolopoulos, N. P. 49
Papert, S. A. 44
Paquet, T. 6, 23
Parizeau, M. 20, 22
Parkkinen, J. 7
Parmentier, F. 6, 22
Parthasarathy, K. 106
Parui, S. K. 49
Pasquer, L. 33
Pastor, J. A. 49
Pavlidis, T. 23, 24
Pearl, J. 20
Pearlmutter, B. A. 104
Personaz, L. 80
Pfister, M. 111
Phillips, I. 8
Pierrel, J.-M. 1
Pierron, L. 6, 23, 24
Pineda, F. J. 105, 106
Pittman, J. A. 34
Pitts, W. 143
Piwowarski, B. 1
Pottier, L. 6

Quinlan, J. R. 20
Quint, V. 1, 8

Rabiner, L. R. 49
Rahman, A. F. R. 1, 50
Ramel, J.-Y. 6
Rangoni, Y. 1, 55, 64, 67, 70, 93
Rastle, K. 35
Raveendran, P. 49
Rebolho, D. C. 111
Reicher, G. M. 33
Rice, S. V. 56, 63
Rigamonti, M. 1
Rigoll, G. 6
Rivals, I. 80
Rivlin, E. 6, 34
Robadey, L. 19, 20
Rodrigues de Souza, L. d. F. 111
Rohwer, R. 105
Rojas, R. 111
Rondel, N. 49
Rosenberg, C. R. 111
Rosenblatt, F. 43–45, 143
Rosenfeld, A. 20, 24, 25
Ross, D. 8
Rumelhart, D. E. 2, 31, 37, 38, 40, 45, 50, 53, 71, 118, 122, 123

Sabourin, R. 79
Säckinger, E. 111
Said, J. N. 10
Sainz Palmero, G. I. 20, 22
Saitoh, T. 15, 16, 26, 27
Sako, H. 50
Samuelides, M. 43
Sankar, A. 120
Saon, G. 50
Sarkar, P. 56
Sato, A. 23, 24
Sawai, H. 111
Schapire, R. E. 66
Schenkel, M. 111
Schmidt, W. 22, 24
Schölkopf, B. 79
Schwartz, J. H. 143
Segui, J. 32
Sejnowski, T. J. 111
Semeraro, G. 19, 20, 22, 24
Seth, S. 14, 16–18, 23, 24, 26, 27, 50, 117
Shah, R. 111

Sharma, S. 111
Sheng, J. Q. 6, 50
Shin, C. 8
Shinogi, T. 6
Simard, P. 111
Singh, R. 49
Siu, W.-C. 50
Smigiel, E. 49
Smith, L. A. 80
Sánchez, G. 6
Snoussi Maddouri, S. 2, 41–43, 53, 71, 74, 113, 123
Soda, G. 6, 28, 49, 50
Solla, S. A. 74, 79
Song, M. 79
Souafi-Bensafi, S. 20, 22
Sperduti, A. 6, 120–122, 125
Spitz, A. L. 49
Srihari, S. N. 9, 12, 13, 16, 23, 24, 26, 27, 50
Starita, A. 120–122, 125
Steinherz, T. 34
Stoppiglia, H. 79, 80
Stork, D. G. 74
Strathy, N. W. 50
Strouthopoulos, C. 49
Stubberud, P. 49
Suen, C. Y. 2, 10, 31, 50, 79
Suenaga, Y. 6
Sugie, N. 6
Sugiyama, M. 111
Sumiya, T. 22, 24
Summers, K. 6, 11–13, 26, 27
Sun, Z. 83, 100
Szilas, N. 107

Tachikawa, M. 15, 16, 26, 27
Taft, M. 33
Takahashi, H. 13, 16, 26, 27
Takahashi, M. 23, 24
Takahashi, T. 6
Tang, Y. Y. 10, 22, 24
Tarnikova, Y. 1
Tateisi, Y. 17, 18, 26, 27
Taylor, I. 33
Taylor, M. M. 33
Tetko, I. V. 48
Thiria, S. 43
Thoma, G. R. 11–13, 21, 22, 26, 27, 71
Thomason, M. G. 19
Tipping, M. 79
Todoran, L. 20, 22
Tokuyasu, T. A. 49
Torkkola, K. 76
Toussaint, Y. 6
Toyokawa, K. 13, 16, 26, 27
Toyozumi, K. 6
Trupin, E. 6, 19, 20
Tsujimoto, S. 13, 16, 26, 27
Tsuruoka, S. 6
Turcan, I. 1
Turolla, E. 6

Vajda, S. 64
Valveny, E. 6
Valverde, N. 6, 23, 24
Vapnik, V. N. 79
Vassilvitskii, S. 97
Vatton, I. 1
Villanueva, J. J. 17
Vincent, L. 63
Vincent, N. 6
Vinciarelli, A. 6
Visa, A. 7
Viswanathan, M. 16–18, 23, 24, 26, 27, 50, 117

Wahl, F. M. 23, 24
Waibel, A. H. 107, 111
Walischewski, H. 20, 22
Wan, E. A. 107, 108, 110
Wang, D. 23, 24
Wang, J. 50
Wang, P. S.-P. 50
Wang, Y.-C. 19, 20
Watanabe, T. 6
Wechsler, H. 21, 22
Weindorf, M. 7
Weiss, I. 6
Wenyin, L. 6
Wenzel, C. 15, 16
Werbos, P. J. 45
Weston, J. 79
Whichello, A. P. 49
Wöhler, C. 111
Widrow, B. 44
Wilcox, C. 1
Wilfong, G. T. 6
Williams, R. J. 45, 106, 120
Wilson, R. C. 11
Wolf, M. 22, 24
Wong, K. W. 49
Wong, K. Y. 23, 24
Worring, M. 20, 22

Xi, D. 22, 24
Xu, L. 91
Xu, Q. 50

Yamaai, T. 15, 16, 26, 27
Yamada, N. 6
Yamashita, A. 13, 16, 26, 27
Yan, H. 49
Yanadume, S. 82

Yang, W.-J. 23, 24, 26, 27
Yanikoglu, B. A. 63
Yoshikawa, T. 6
You, D. 50
Yu, B. 23, 24
Yu, L. 80

Zack, G. W. 9
Zanibbi, R. 6
Zhang, X.-W. 6
Zhong, M. 111
Zhong, Y. 49
Zhou, J. 23, 24, 50
Zipser, D. 106, 120
Zramdini, A. 18, 20, 26, 27
Zucker, S. W. 20

Index

accélération de la reconnaissance 74
agent 22
algorithme génétique 79
ALTO 8, 55, 70
ambiguïté 25
analyse d'images de documents 5
analyse en composantes principales 80, 87
analyse physique de documents 5
analyse syntaxique 17
analyse, RNP 62
apprentissage incrémental 19
apprentissage neuronal flou 20
apprentissage par cœur 45
approche ascendante 22
approche descendante 22
approche dirigée par le modèle 10, 11
approche dirigée par les données 10, 20
approximation de fonction 45
arbre de décision 14
arbre géométrique 19

base de connaissance 12
binarisation 49
bloc, boîte englobante 64
boosting 66
branch and bound 79

carte auto-organisatrice 49
cas d'arrêt 28
Cattell 92
champs de Markov 20
classement de variables 76
classification 45
classification de pixel 49
codage des données 54
coefficient de corrélation 76
combinaison de classifieur 50
comparaison de graphes 17
complexité, PMC 73
composante principale 82
compréhension de documents 6
contexte 33, 35, 56
convergence, PMC 28, 47
correction 63
correction d'inclinaison 49
critère d'arrêt, PMC 47
critère de Fisher 76
cycle perceptif 38, 39, 61, 65

décomposition en ondelettes 22
décomposition en valeurs singulières 81
DAFS 8
descripteur de Fourier 42
document de vérité 24, 59
document graphique 6
document manuscrit 6
document structuré 6
DTD 9, 55
dual route 35

échantillon type 63
écriture en ligne 33
entropie de Shannon 23
équation d'activation 39
évaluation des performances 24

fonction seuil 43

gradient, descente 46
grammaire bidimensionnelle 16
grammaire EBNF 16
grammaire formelle 16

HMM 49
hypothèse, génération, validation, insertion 40

indice d'homogénéité 22
inférence de grammaire d'arbre 18
information contextuelle 35
information linguistique 32
information mutuelle 76
interprétation de documents 6
interprétation multicontextuelle 33
intervalle d'influence 15

Kaiser 92
k-means 97

ligne de temporisation 111, 113
logique des prédicats 22
logique floue 14

logogène 33

mécanisme de lecture 33
méthode à adaptateur 78
méthode embarquée 76, 78, 79
méthode inductive 19
méthode par filtre 76, 78, 79
métrique de comparaison 24
macrostructure 6
METS 9, 70
microstructure 6
modèle cognitif de lecture 32
modèle de graphe 19
modèle formel de document 25
modèle générique 19
multirelationnel 21

n-grammaire 18
neurone complexe récursif 120
neurone formel 43
niveau d'interprétation 31

OCR 55
ODA 8
optimal brain damage 74
optimal brain surgeon 74
ordre des connexions 108

partitionnement de l'espace d'entrée 83, 85, 87
pas d'apprentissage 46
Perceptron multicouche 21, 43, 44
point fixe 23
précision 18, 24
prétraitement de l'image 49
primitive physique 53
propagation, RNP 61

réduction de données 75, 80
réduction du bruit 49
région d'intérêt 23
réseau à décalage temporel 104, 107
réseau à retour d'état 106
réseau à retour de contexte 106
réseau bouclé 104
réseau de concepts 22
réseau de Hopfield 106
réseau de neurones dynamique perceptif 112
réseau de neurones perceptif 59
réseau de neurones récurrents 119
réseau de neurones transparent 42
réseau dynamique 104
réseau dynamique, convergence 107
réseau récurrent temporel 106
réseau statique récurrent 104
réseaux bayésiens 20
rétro-conversion 6
rétropropagation dans le temps 106
rétropropagation du gradient 45
rétropropagation récurrente 105, 106
rétropropagation temporelle 109
rappel 18, 24
reconnaissance de caractères 50
reconnaissance de documents 5
registre à décalage 111
rejet 24, 25
relaxation 20
relaxation (méthode de) 13
représentation distribuée 35
représentation en arbre 12
représentation locale 34
restauration de texte 49

sélection de données 75
sélection de sous-ensembles de variables 75, 78
schéma multiagent 22
script 12
segmentation de caractères 50
segmentation de pages 23, 49
segmentation de régions 49
segmentation implicite 34, 41
SGML 8
Siggraph 67
sigmoïde 44
solveur de contraintes 12
structure logique 7, 9, 56
structure physique 7, 9, 53
suppression de ligne 49
sur-segmentation 64
surapprentissage 48
système à base de règles 11

tableau noir 15, 33
TEI 8, 56, 70
traitement du signal 49

valeur propre 82
validation croisée 48
variance expliquée 82, 91
vecteur propre 82
verification model 35
vision globale, vision locale 86

XML 70
XSLT 70
XY-cut 14, 23

Résumé

L'extraction de structures logiques de documents est un défi du fait de leur complexité inhérente et du fossé existant entre les observations extraites de l'image et leur interprétation logique. La majorité des approches proposées par la littérature sont dirigées par le modèle et ne proposent pas de solution générique pour des documents complexes et bruités. Il n'y a pas de modélisation ni d'explication sur les liens permettant de mettre en relation les blocs physiques et les étiquettes logiques correspondantes. L'objectif de la thèse est de développer une méthode hybride, à la fois dirigée par les données et par le modèle appris, capable d'apprentissage et de simuler la perception humaine pour effectuer la tâche de reconnaissance logique. Nous avons proposé le Réseau de Neurones Dynamique Perceptif qui permet de s'affranchir des principales limitations rencontrées dans les précédentes approches. Quatre points principaux ont été développés :

- utilisation d'une architecture neuronale basée sur une représentation locale permettant d'intégrer de la connaissance à l'intérieur du réseau. La décomposition de l'interprétation est dépliée à travers les couches du réseau et un apprentissage a été proposé pour déterminer l'intensité des liaisons ;
- des cycles perceptifs, composés de processus ascendants et descendants, accomplissent la reconnaissance. Le réseau est capable de générer des hypothèses, de les valider et de détecter les formes ambiguës. Un retour de contexte est utilisé pour corriger les entrées et améliorer la reconnaissance ;
- un partitionnement de l'espace d'entrée accélérant la reconnaissance. Des sous-ensembles de variables sont créés automatiquement pour alimenter progressivement le réseau afin d'adapter la quantité de travail à fournir en fonction de la complexité de la forme à reconnaître ;
- l'intégration de la composante temporelle dans le réseau permettant l'intégration de l'information de correction pendant l'apprentissage afin de réaliser une reconnaissance plus adéquate. L'utilisation d'un réseau à décalage temporel permet de tenir compte de la variation des entrées après chaque cycle perceptif tout en ayant un fonctionnement très proche de la version statique.

Mots-clés : analyse d'images de documents, réseau de neurones transparent, Perceptron multicouche, réseau de neurones dynamique, cycle perceptif, correction et sélection de variables

Abstract

Logical structure extraction of documents remains a challenging problem due to their inherent complexity and the gap between the physical features extracted from the image and their corresponding logical interpretation. Most of the literature approaches propose model-driven approaches which are not generic enough to handle complex and noisy documents. They do not use intermediate interpretation steps and do not explain the relationships between the physical blocks and the corresponding logical labels. The main objective of this thesis is to develop a hybrid method, using both data-driven and model-driven approach, which is capable to learn the relationships and simulate human perception during the logical recognition task. We have proposed a Dynamic Perceptive Neural Network which can handle drawbacks of previous systems. Four main points have been developed:

- a special network topology based on local representation where the knowledge can be integrated. The logical interpretation is unfolded along the layers of the network and a training stage is performed to find the weights for each link;
- perceptive cycles (several bottom-up and top-down processes) perform the recognition. The network is able to generate hypothesis, validate them and detect ambiguous patterns. The context manages the correction of the input features to improve the recognition rate;
- an input feature clustering has been proposed to speed-up the recognition. Subsets of features are automatically computed and are given progressively to feed the network in order to adapt the amount of computations according to the pattern complexity;
- dynamic integration in the network that make it possible to integrate the data correction information during the training stage to have more appropriate behavior during the recognition. The improvement uses a Time Delay Neural Network architecture to take into account the input data variations after each perceptive cycle while the recognition step is quite similar to the static one.

Keywords: document image analysis, transparent neural network, multilayer Perceptron, dynamic neural network, perceptive cycle, feature correction and selection