

UNIVERSITE ABOU BEKR BELKAID TLEMCEN
Faculté des sciences de l'ingénieur
Département de télécommunication



MEMOIRE

Pour l'obtention du diplôme de magistère SRT
(Systèmes et Réseaux de Télécommunications)
Option : Réseaux

Présentée et soutenue publiquement
Par :

Réda BOUABDALLAH

=====
Raisonnement à partir de cas :
Une technique d'adaptation des cas basée sur les réseaux de
neurones
=====

Dirigé par : Mr CHIKH Azeddine

JURY

Mr FEHAM Mohamed	Professeur à l'université de Tlemcen	Président
Mr CHIKH M ^{ed} Amine	M.C à l'université de Tlemcen	Examineur
Mr ZIANI CHERIF Salim	C. C à l'université de Tlemcen	Examineur
Mr CHIKH Azeddine	M.C à l'université de Tlemcen	Directeur

2006-2007

Remerciements :

Tout d'abord, Je tiens à remercier mon dieu tout puissant pour l'aide et la santé qui m'a donné pour travailler.

Je remercie également mon directeur de ce mémoire, Mr CHIKH Azeddine pour avoir accepté de diriger ce travail et pour avoir suivi mes travaux durant mon mémoire.

Je remercie aussi les membres du jury, Messieurs Feham Mohamed, Chikh Amine et Ziani Chrif Salim pour avoir accepté d'évaluer le présent travail.

Je témoigne ma gratitude à toute ma famille qui n'a pas cessé de m'encourager et de me soutenir durant toute ma vie.

Le souvenir du passé nous donne une certaine sagesse pour affirmer nos pas dans l'avenir. Permettez-moi, enfin, de rendre hommage à mon père décédé, dont le souvenir me permet d'avoir espoir en l'avenir.

SOMMAIRE

RESUME	9
INTRODUCTION GENERALE	10
1. INTRODUCTION	13
2. HISTOIRE DE L'IA	13
3. QUELQUES DEFINITIONS DE L'IA	14
4. COMBAT DEEP BLUE/KASPAROV	14
5. APPROCHES DE L'IA	15
5.1. APPROCHE COGNITIVE (PENSER COMME DES HUMAINS)	15
5.2. APPROCHE PRAGMATISTE (AGIR COMME DES HUMAINS)	15
5.3. APPROCHE CONNEXIONNISTE (NUMERIQUE)	16
6. SYSTEME SYMBOLIQUE PHYSIQUE	16
7. QUELQUES TECHNIQUES DE L'IA	17
7.1. SYSTEME A BASE DE REGLES	17
7.1.1. Définition	17
7.1.2. Composants d'un système expert	17
7.1.3. Quelques exemples de systèmes experts	18
7.2. ALGORITHMES GENETIQUES	18
7.2.1. Présentation	18
7.2.2. Les opérations génétiques	19
7.2.3. Algorithme	20
7.2.4. Inconvénients :	20
7.3. AGENTS INTELLIGENTS OU KNOWBOT	20
7.3.1. Qu'est ce qu'un agent ?	20
7.3.2. Caractéristiques d'un agent	21
7.3.3. Système multi-agents	21
7.3.4. Types d'agent	21
7.4. APPRENTISSAGE PAR RENFORCEMENT	22
7.4.1. Définition	22
7.4.2. Composant d'un système d'apprentissage par renforcement	22
– Exemple	22
7.5. ARBRES DE DECISION	23
7.5.1. Composants d'un arbre de décision	23
7.5.2. Fonctionnement	23
7.6. LOGIQUE FLOUE	23
7.6.1. Principe	23
7.6.2. Fonctionnement	24
7.7. LE DATA MINING	25
7.7.1. Définition	25
7.7.2. Les tâches du Data Mining	25
7.8. RESEAUX BAYESIENS	25
7.8.1. Principe	25
7.8.2. Fonctionnement	26
7.8.3. Mise en place	26
7.9. RAISONNEMENT A PARTIR DE CAS	26
7.9.1. Principe	26
7.9.2. Processus RÀPC	26
7.10. RESEAUX DE NEURONES	26
7.10.1. Présentation	26
7.10.2. Processus du raisonnement	27
8. QUELQUES DIFFICULTES RENCONTREES EN IA (LIMITES)	27
8.1. COMPLEXITE	27
8.2. EXPLOSION COMBINATOIRE	27

9. LES DOMAINES CONCERNES PAR L'IA	28
9.1. COMPREHENSION OU TRAITEMENT DU LANGAGE NATUREL (NLP OU TLN)	28
9.2. ENSEIGNEMENT A DISTANCE (E-LEARNING)	28
9.3. DEMONSTRATION ET DECOUVERTE AUTOMATIQUE DE THEOREMES	28
9.4. ROBOTIQUE	28
9.5. PROGRAMMATION AUTOMATIQUE ET VERIFICATION DE PROGRAMMES	28
9.6. AUTRES DOMAINES	28
10. LES PRINCIPAUX LANGAGES DE L'INTELLIGENCE ARTIFICIELLE	ERREUR !
SIGNET NON DEFINI.	
11. CONCLUSION	30
1. INTRODUCTION	32
2. DEFINITION	32
3. QUAND UTILISE T'ON UN SYSTEME DE RAPC ?	32
4. TYPOLOGIE DE CONNAISSANCES	33
4.1. LES CAS	33
4.2. LES CONNAISSANCES DU DOMAINE (VOCABULAIRE D'INDEXATION)	33
4.3. LES CONNAISSANCES DE SIMILARITE	33
4.4. LES CONNAISSANCES D'ADAPTATION	33
5. PROCESSUS DE RAISONNEMENT A PARTIR DE CAS	34
5.1. ÉLABORATION	34
5.2. REMEMORATION	34
5.3. ADAPTATION	35
5.3.1. <i>La dimension produit (transformationnelle, structurelle)</i>	35
5.3.2. <i>La dimension processus (dérivationnelle, générative)</i>	36
5.4. REVISION	36
5.5. APPRENTISSAGE	36
6. MODELES DE REPRESENTATION DE CAS	37
6.1. MODELE STRUCTUREL	37
6.2. MODELE CONVERSATIONNEL	37
6.3. MODELE TEXTUEL	38
7. MODELES D'ORGANISATION DE LA BASE DE CAS	39
7.1. MODELE SIMPLE	39
7.2. MODELE A MEMOIRE DYNAMIQUE	40
7.3. MODELE A BASE DE CATEGORIES	41
8. QUELQUES PROJETS EN RAPC	42
9. QUELQUES RENCONTRES SUR LE RAPC	44
10. CONCLUSION	45
1. INTRODUCTION	47
2. HISTORIQUE	47
3. DU NEURONE BIOLOGIQUE AU NEURONE FORMEL	47
3.1. NEURONE BIOLOGIQUE	48
3.1.1. <i>Composants d'un neurone</i>	48
3.1.2. <i>Fonctionnement</i>	48
3.1.3. <i>Notion de synapse</i>	49
3.1.4. <i>La transmission au niveau du synapse</i>	49
3.2. NEURONE FORMEL	50
3.2.1. <i>Description</i>	50
3.2.2. <i>Fonctionnement</i>	50
3.2.3. <i>Fonction d'activation</i>	51

4. RESEAUX DE NEURONES	53
5. ARCHITECTURES DES RESEAUX DE NEURONES	54
5.1. ARCHITECTURE EN COUCHES (FEED-FORWARD)	54
5.2. ARCHITECTURE RECURRENTTE (FEEDBACK).....	54
6. APPRENTISSAGE DES RESEAUX DE NEURONES	55
6.1. DEFINITION.....	55
6.2. APPRENTISSAGE SUPERVISE.....	55
6.3. APPRENTISSAGE NON SUPERVISE (AUTO ORGANISATIONNEL).....	55
6.4. APPRENTISSAGE SEMI-SUPERVISE	56
6.5. APPRENTISSAGE PAR RENFORCEMENT	56
7. MISE EN OEUVRE D’UN RESEAU DE NEURONES.....	56
7.1. PREPARATION DES DONNEES.....	56
7.2. DETERMINATION DES PARAMETRES	56
7.3. PHASE D’APPRENTISSAGE	57
7.4. PHASE DE TEST	57
7.5. PHASE D’EXPLOITATION	57
8. QUELQUES RESEAUX CELEBRES.....	57
8.1. LE PERCEPTRON	57
8.2. ADALINE.....	58
8.3. LES PERCEPTRONS MULTICOUCHES (PMC OU MLP).....	59
8.3.1. <i>Présentation</i>	59
8.3.2. <i>Algorithme de retro-propagation</i>	59
8.3.3. <i>Règle d’apprentissage</i>	60
8.3.4. <i>Les applications du PMC</i>	60
8.4. LES RESEAUX RBF (RADIAL BASIC FUNCTION)	61
8.5. LES RESEAUX DE HOPFIELD.....	61
8.5.1. <i>Description</i>	61
8.5.2. <i>Dynamique du réseau</i>	62
8.5.3. <i>Fonction d’énergie</i>	62
8.6. RESEAUX DE KOHONEN.....	62
8.6.1. <i>Dynamique du réseau</i>	64
8.6.2. <i>Apprentissage</i> :	64
9. DOMAINES D’APPLICATION	64
10. AVANTAGES ET INCONVENIENTS	64
10.1. AUTRES AVANTAGES	65
10.2. INCONVENIENTS.....	65
11. CONCLUSION	66
1. INTRODUCTION	68
2. QUELQUES TECHNIQUES D’ADAPTATION EN RAPC	68
2.1. [HAMMOND 1989]	68
2.2. [HASTINGS ET AL., 1995]	68
2.3. [HANNEY ET AL., 1996]	69
2.4. [SMYTH 95; SMYTH ET AL., 1998].....	69
2.5. [LEAKE ET AL., 1997]	69
2.6. [WILKE 1997].....	70
2.7. [PURVIS ET AL., 1998].....	70
2.8. [BERGMANN ET AL., 1995 ; BERGMANN ET AL., 1996].....	70
2.9. [ANAND ET AL., 1998]	70
2.10. [DE SILVA GARZA ET AL., 1999]	71
2.11. [CORCHADO ET AL., 2001]	71
2.12. [JARMULAK ET AL., 2001].....	71
2.13. [LIEBER ET AL., 2001; LIEBER ET AL., 2003].....	72

2.14. [D'AQUIN ET AL., 2004]	72
2.15. AUTRES TRAVAUX	72
3. COMPARAISON ENTRE CES TRAVAUX	72
3.1. NATURE DE LA TECHNIQUE	73
3.2. TYPE DE CONNAISSANCES ACQUISES	74
4. CONCLUSION	75
1. INTRODUCTION	76
2. TECHNIQUE PROPOSEE	77
2.1. CREATION DES GDS	77
2.2. SELECTION DES CAS PIVOTS	77
2.3. APPRENTISSAGE DES RESEAUX DE NEURONES	78
2.4. PROCESSUS DU RAISONNEMENT	79
3. CAS D'APPLICATION : ADAPTATION DE SCENARIOS PEDAGOGIQUES	80
3.1. PRESENTATION DU MODELE IMS LIP	80
3.2. ADAPTATION DES SCENARIOS D'APPRENTISSAGE PAR LA TECHNIQUE PROPOSEE	84
3.2.1. Saisie des cas	85
3.2.2. Création des GDS	86
3.2.3. Sélection de cas pivots	87
3.2.4. Paramètres des réseaux de neurones	87
• Codage des entrées (caractéristiques des apprenants)	87
• Codage des sorties (scénarios d'apprentissage)	87
• Couches cachets	88
3.2.5. Processus de raisonnement	88
4. COMPARAISON DE LA TECHNIQUE PROPOSEE AVEC CELLE DE [CORCHADO 2001]	89
5. CONCLUSION	90
<i>CONCLUSION GENERALE</i>	91
1 INTRODUCTION	92
2 L'ADAPTATION EN RAPC	93
3 TECHNIQUES D'ADAPTATION ET D'ACQUISITION DE CONNAISSANCES D'ADAPTATION	94
4 TECHNIQUE PROPOSEE	95
4.1 CREATION DES GDS	95
4.2 SELECTION DES CAS PIVOTS	95
4.3 APPRENTISSAGE DES RESEAUX DE NEURONES	96
4.4 PROCESSUS DU RAISONNEMENT	97
5 CAS D'APPLICATION	98
6 CONCLUSION	100
7 BIBLIOGRAPHIE	100
BIBLIOGRAPHIE	102

Liste des Figures :

Figure 1- 1: Test de Turing.	16
Figure 1- 2 : l'opération de mutation	19
Figure 1- 3: l'opération de croisement	19
Figure 1- 4: l'opération d'inversion	20
Figure 1- 5 : exemple comparatif entre logique booléenne/floue	24
Figure 1- 6: processus d'un système flou.	24
Figure 2- 1: Cycle RàPC	34
Figure 2- 2 : Schéma d'adaptation.	35
Figure 2- 3: représentation hiérarchique de prototypes combinée avec les arbres de décision	40
Figure 2- 4 : Représentation par le modèle à mémoire dynamique.	41
Figure 2- 5: Représentation à base de catégories	41
Figure 3- 1: Neurone biologique extrait de [Hérault et al., 1994]	48
Figure 3- 2: Les synapses [Haykin 1994].	49
Figure 3- 3: Fonctionnement d'une synapse [Haykin 1994].	49
Figure 3- 4: Exemple du neurone formel.	50
Figure 3- 5 : Neurone formel [Wikipédia].	51
Figure 3- 6 : Fonctions Heaviside (à seuil)	52
Figure 3- 7 : Fonctions signe	52
Figure 3- 8 : Fonction linéaire.	52
Figure 3- 9 : Fonction à saturation.	52
Figure 3- 10 : Fonction sigmoïde	53
Figure 3- 11 : Exemple d'un réseau de neurones	53
Figure 3- 12 : Réseaux feed-forward.	54
Figure 3- 13 : Réseaux feedback.	55
Figure 3- 14 : Le Perceptron de Rosenblatt [Gosselin 1996]	58
Figure 3- 15 : Perceptron multicouches	59
Figure 3- 16: Le réseau RBF	61
Figure 3- 17: Architecture du réseau de Hopfield.	62
Figure 5-1 : Décomposition de la base de cas	77
Figure 5- 2 : Fonction d'adaptation	78
Figure 5- 3 : Notre processus RàPC	79
Figure 5- 4 : Catégories d'IMS LIP [IMS].	83
Figure 5- 5 : la fenêtre de saisie.	86
Figure 5- 6 : Fenêtre pour la création de GDS	86
Figure 5- 7 : fenêtre d'adaptation	88

Liste des Tableaux :

Tableau 2- 1: Un cas voiture représenté par le modèle structurel.....	37
Tableau 4- 1 : Comparaison selon la nature et la source de connaissance	74
Tableau 5-1 : Catégorie <i>Identification</i> (adapté de [Razmerita, 2003]).	81
Tableau 5-2 : Catégorie <i>Goal</i> [Razmerita, 2003].....	81
Tableau 5-3 : Catégorie <i>QCL</i> [Razmerita, 2003].....	81
Tableau 5-4 : Catégorie <i>Activity</i> (adapté de [Razmerita, 2003]).	81
Tableau 5-5 : Catégorie <i>Interst</i> (adapté de [Razmerita, 2003]).	82
Tableau 5-6 : Catégorie <i>Competency</i> (adapté de [Razmerita, 2003]).	82
Tableau 5-7 : Catégorie <i>Accessibility</i> [Razmerita, 2003].	82
Tableau 5- 8 : Catégorie <i>Affiliation</i> (adapté de [Razmerita, 2003]).	83
Tableau 5- 9 : Catégorie <i>SecurityKey</i> (adapté de [Razmerita, 2003]).	83
Tableau 5-10 : Exemples d'apprenants.....	84
Tableau 5- 11 : Exemples de scénarios d'apprentissage.....	85

Liste des Abréviations:

ACA : Acquisition de Connaissances d'Adaptation.
CBR : Case Based Reasoning.
ECD : Extraction de Connaissances à partir de Données.
GDS : Groupe De Similarité.
IA : Intelligence Artificielle.
IMS : Information Management System.
LIP : Learner Information Package.
PMC : Perceptron Multi-Couches.
RàPC : Raisonnement à Partir de Cas.
RN : Réseaux de Neurones.
SMA : Système Multi-Agents.
TLN : Traitement du Langage Naturel.

Résumé

Le raisonnement à partir de cas (RàPC) est un paradigme issu de l'intelligence artificielle qui permet la mise en place d'un système à base de connaissances. Il consiste à résoudre un nouveau problème en se remémorant et en adaptant un ou plusieurs cas passés déjà résolus. L'adaptation constitue l'étape la plus importante du processus de RàPC. Elle nécessite l'acquisition d'un ensemble de connaissances spécifiques généralement très difficiles à modéliser. Dans ce mémoire, nous proposons une technique d'adaptation des cas qui est basée sur l'utilisation de réseau de neurones. On fournit à l'entrée du réseau le nouveau problème rencontré et on récupère à sa sortie la solution adaptée. Cette technique présente deux principaux avantages: (1) l'apprentissage des réseaux de neurones se fait avant le processus de raisonnement; (2) les réseaux générés peuvent être réutilisés dans les futures adaptations. Elle sera appliquée à l'adaptation des scénarios d'apprentissage pédagogique, décrits sous forme de suites d'activités, aux apprenants décrits avec le modèle IMS-LIP.

Mots clés: Raisonnement à partir de cas (RàPC), adaptation, acquisition de connaissances d'adaptation (ACA), réseaux de neurones, scénario d'apprentissage, apprenant.

Abstract

Case-based reasoning (CBR) is a paradigm of Artificial Intelligence that allows setting-up of a knowledge-based system. It tries to resolve a new problem by retrieving and adapting past cases which are already resolved. The adaptation constitutes the most important step of the CBR process. It requires the acquirement of a set of specific knowledge which is very difficult to model. In this thesis, we present a case adaptation technique which is based on neural networks. We present at the input of the network the new problem and we recover at the output the adapted solution. This technique presents two main advantages: (1) the learning of the neural network is realized before the CBR process; (2) the generated networks can be reused in the future adaptations. This technique will be applied to the learning scenarios adaptation, described with sequence of activities, to the learners described with IMS-LIP.

Keywords: Case based reasoning (CBR), adaptation, adaptation knowledge acquisition (AKA), neural networks, learning scenario, learner.

Introduction générale

L'avènement des technologies de l'information et de communication a favorisé le partage d'informations, d'expériences et du savoir-faire entre les différents acteurs d'un système informatique.

Pour résoudre un nouveau problème, on fait souvent appel et on essaye d'abord des méthodes qui avaient bien fonctionné dans des anciens problèmes similaires.

Les expériences et les savoir-faire des individus forment un capital très important pour tous les systèmes, d'où la nécessité d'utiliser des techniques de l'ingénierie de connaissances et de l'intelligence artificielle qui permettent leur capitalisation, leur gestion, leur partage et leur réutilisation.

Le raisonnement à partir de cas est l'une de ces techniques. Son principe est de capitaliser les expériences pertinentes vécues et déjà résolues (problèmes et leur solutions) pour ensuite, les utiliser pour résoudre des problèmes similaires tout en adaptant les anciennes solutions aux nouveaux problèmes.

Organisation de ce mémoire

Ce mémoire comporte 5 chapitres et elle est organisée comme suite :

- Le premier chapitre est une introduction du domaine de l'intelligence artificielle. Tout d'abord, un historique de l'intelligence artificielle sera présenté suivi de quelques définitions. Ensuite, nous citons les différentes approches de l'intelligence artificielle. Nous présentons aussi dans ce chapitre quelques techniques de l'intelligence artificielle et les difficultés rencontrées lors de l'implémentation de ces techniques. Finalement, nous citons quelques domaines d'application de l'intelligence artificielle.
- Le deuxième chapitre sera consacré pour présenter la technique du raisonnement à partir de cas. Tout d'abord nous définissons ce paradigme de l'intelligence artificielle. Ensuite, nous citons quelques conditions pour qu'un système de raisonnement à partir de cas soit efficace. Nous rappelons aussi les différents types de connaissances utilisées par un système de raisonnement à partir de cas. Après, nous citons les différentes étapes du cycle de raisonnement à partir de cas. Nous présentons aussi les modèles de représentation de cas et les modèles d'organisation de la base de cas.
- Dans le troisième chapitre, nous présentons une autre technique de l'intelligence artificielle (Réseaux de neurones). Nous commençons par un historique suivi d'une description de la base du neurone formel (neurone biologique). Nous évoquons quelques définitions des réseaux de neurones avec quelques architectures. Nous présentons aussi quelques réseaux de neurones célèbres avec les domaines où cette technique est très appliquée.
- L'avant dernier chapitre comporte un état de l'art des techniques d'adaptation et d'acquisition de connaissances d'adaptation. Nous

présentons pour chaque technique son principe et son domaine d'application. Ensuite, une comparaison entre ces techniques selon quelques caractéristiques sera montrée.

- Dans le dernier chapitre, nous présentons tout d'abord notre contribution qui consiste à proposer une nouvelle technique d'adaptation et d'acquisition de connaissances associées. Après, nous décrivons le domaine où nous allons tester cette technique (adaptation des scénarios pédagogiques). Finalement, nous présentons l'application de cette technique dans ce domaine d'application.

Chapitre I

Intelligence artificielle

1. Introduction

Notre travail de recherche s'inscrit dans le cadre d'une technique de l'intelligence artificielle (IA) qui est le raisonnement à partir de cas (RàPC). Cette technique, nous allons la couplée avec une autre technique qui est les réseaux de neurones.

Dans ce chapitre, nous allons présenter le domaine de l'IA, pour ensuite (dans les chapitres deux et trois), se spécialiser sur les deux techniques citées ci-dessus.

Depuis très longtemps, les humains tentent de concevoir des machines qui soient capables de réaliser des tâches à leur place et surtout celles qui sont risquées et que les humains ne peuvent pas les faire, comme par exemple l'étude des volcans, la fabrication microscopique, l'espionnage dans des guerres avec des avions sans pilote, l'exploration spatiale...etc.

L'avènement des technologies de l'information et de communication a poussé les chercheurs à travailler plus et à croire à cette nouvelle tendance qui vient de s'imposer jour après jour avec les résultats qui se réalisent dans plusieurs domaines d'application tels que la médecine, la traduction automatique, la recherche d'information, la robotique, la programmation des jeux...etc.

L'intelligence artificielle (IA) est une branche (discipline, filière) de l'informatique qui s'intéresse à rendre les machines (Pc, Robot...) plus au moins autonome pour quelles puissent prendre des décisions et tirer des conclusions sans l'intervention d'un opérateur humain. Elle vise à créer des systèmes capables de reproduire un comportement similaire à celui des humains face à des situations de raisonnement. L'objectif principal de l'IA est l'automatisation des tâches.

Ce chapitre va être organisé comme suite : nous allons, tout d'abord présenter un historique de l'IA. Puis, nous rappelons quelques définitions de l'IA (section 3). La section 4 sera consacrée au combat Deep-Blue/Kasparov qui a prouvé que ce domaine est prometteur. Dans la section 5, nous décrirons les différentes approches de l'IA. Après, nous introduisons la notion de système symbolique physique (section 6). Dans la section 7, nous citons quelques techniques utilisées dans l'IA. Ensuite (section 8), nous évoquons les difficultés rencontrées dans l'implémentation de ces techniques (limites). Finalement, dans les sections 9 et 10, nous citons respectivement quelques domaines d'application de l'IA et les langages les plus utilisés dans ce domaine.

2. Histoire de l'IA

Une première tentative de créer un système intelligent été de construire un traducteur automatique, mais cette dernière a échoué car les concepteurs n'ont pas tenu compte de la notion de sémantique. Cet échec a permis de tirer des enseignements à savoir :

- Importance des connaissances non exprimées,
- Ouverture d'un domaine de recherche sur la représentation des connaissances,
- Impossibilité d'exprimer toutes les connaissances d'un individu,
- Nécessité d'exprimer les connaissances sous forme générique.

Le terme « *intelligence artificielle* », souvent abrégée avec le sigle *IA*, a été utilisé pour la première fois par McCarthy dans sa rencontre avec Newell, Minsky et Simon

au collège de darmouth (New hampshire, USA) en 1956. Au cours de cette année, Newell, Show et Simon, ont créé le premier programme en IA qui est nommé « *LOGIC THEORIST* » dont le but est de permettre la démonstration automatique de théorèmes mathématiques. Ce programme a permis la découverte d'un nouveau théorème mathématique et il a démontré un théorème, avec une preuve plus courte que dans le livre.

- Un an après (1957), ils ont développé un programme de résolution de problème généraux (GPS : General Problem Solver) et un autre de jeu d'échecs (NSS).
- En 1960, McCarthy a crée le langage *Lisp* pour traiter les données symboliques non structurées, ce dernier a pris sa place comme langage de l'IA pour les prochaines années.
- En 1965, Joseph Weizenbaum a créé le système ILISA: un système intelligent qui dialogue en anglais et qui joue en psychothérapie
- En 1971, Colmerauer a créé lui aussi le célèbre langage Prolog qui a fait une révélation dans le domaine de l'IA et Terry Winograd a élaboré le système SHRDLU qui est capable de dialoguer et d'agir.
- Dans les années 80, et avec l'arrivée puissante des japonais dans le domaine de l'IA par le lancement du grand projet *Cinquieme génération* dont le but est de développer des logiciels, des matériaux et des techniques pour accélérer les recherches dans le domaine de l'IA.

3. Quelques définitions de l'IA

Plusieurs définitions de l'IA existent dans la littérature, nous citons quelques unes :

« Art de créer des machines qui effectuent des fonctions requérant de l'intelligence lorsqu'elles sont réalisées par des humains ». [Kurzweil 1990]

« Automatisation d'activités associées à la pensée humaine comme la prise de décision, la résolution de problèmes, l'apprentissage, etc. ». [Bellman 1978]

« Étude de la façon de réaliser des ordinateurs capables d'effectuer des opérations présentement les mieux réalisées par des humains ». [Rich et Knight 1991]

4. Combat Deep Blue/Kasparov

En 1997, à Philadelphie, le champion du monde aux échecs, *Garry Kasparov* a été battu par un ordinateur d'IBM nommé *DEEP BLUE* et qui est doté d'un programme intelligent et de 512 processeurs, après six manches de jeu.

Cette machine effectue une recherche sur 14 niveaux et explore plus de 1 milliard de combinaisons par coup à raison de plus de 200 millions de positions par seconde, utilise une fonction d'évaluation très sophistiquée et des méthodes pour étendre certaines recherches à plus de 40 niveaux.

Cette victoire d'une machine dotée d'un programme intelligent face à un champion du monde qui était imbattable a poussée les gents et les chercheurs à croire plus à cette nouvelle tendance qui est dans ses premiers pas.

5. Approches de l'IA

Trois approches de l'IA sont distinguées: approche cognitive, approche pragmatiste et approche connexionniste. Nous décrirons, dans ce qui suit, chacune d'elles.

5.1. Approche cognitive (Penser comme des humains)¹

C'est une approche pluridisciplinaire (sciences humaines, la psychologie...) qui regroupe des informaticiens, linguistes et psychologues. Elle s'intéresse à étudier les processus du raisonnement humain.

William J. Rapaport [Rapaport 1983] la définit ainsi :

« La science cognitive en général cherche à comprendre les fonctions cognitives humaines en termes d'états mentaux et de processus i.e., en termes d'algorithmes qui réalisent la transformation des données d'entrée en données de sortie... L'approche calculatoire de la science cognitive dit que : (1) il existe des états mentaux et des processus qui interviennent entre les stimuli d'entrée et les réponses en sortie, (2) ces états mentaux et ces processus sont des algorithmes (forme forte de la science cognitive) ou en tous cas sont semblables à des algorithmes (forme faible) et, donc, (3) ces états mentaux et ces processus sont susceptibles d'être étudiés scientifiquement ».

L'IA est vue par les défenseurs de cette approche comme « la réalisation des programmes imitant dans leur fonctionnement l'esprit humain » [Alliot et al., 2002]. C'est-à-dire que le processus de raisonnement doit être similaire à celui des être humains. Ici, on s'intéresse à la façon dont les machines effectuent les tâches. L'étude des processus de raisonnement humain est une fin en soi.

Cette approche est utilisée dans la résolution de problèmes complexes qui nécessite l'intervention d'opérateur humain. Un exemple d'utilisation de cette approche est les systèmes de compréhension du langage naturel.

5.2. Approche pragmatiste (Agir comme des humains)

L'objectif dans ce type d'approche est de produire des algorithmes à partir de l'étude du problème tout en tenant compte des contraintes matérielles. Dans cette approche, on ne s'intéresse pas à la façon d'obtenir les résultats, mais on s'intéresse aux résultats eux-mêmes fournis par le système. Le plus important dans cette approche est que le système doit agir comme des humains.

Pour les pragmatistes, un système d'IA est considérée comme une boîte noire qui transforme des entrées en sorties similaires à ceux fournis par un expert humain sans se préoccupe de la manière dont ils seront obtenue. Cette boîte noire sera considérée comme un système intelligent à compter du moment où elle réussirait un certain nombre de tests, qui, réussis par un humain, permettraient de dire qu'il est « intelligent » (au moins pour le domaine concerné). [Alliot et al., 2002]. Le test de Turing se rapproche de cette définition. Il est utilisé pour illustrer cette approche. Nous décrirons ce test ci-dessous :

En 1950, le mathématicien britannique *Alan Turing* publie, dans le journal philosophique *Mind*, un article intitulé *Computing Machinery and Intelligence* où il

¹ Certains, comme Searle, parlent d'IA forte et d'IA faible. Ces termes recouvrent respectivement l'IA cognitive et l'IA pragmatiste.

décrit un test qui a pris son nom (Test de Turing). Ce test consiste à faire communiquer une personne à travers un réseau d'ordinateurs avec deux interlocuteurs invisibles dont l'un est une autre personne et l'autre un système informatique. La personne doit alors deviner lequel des deux interlocuteurs est l'être humain et lequel est système informatique. Si au bout de quelques minutes, elle ne parvient pas à prendre une décision, alors on peut admettre que ce système informatique possède une intelligence (Figure 1-1).

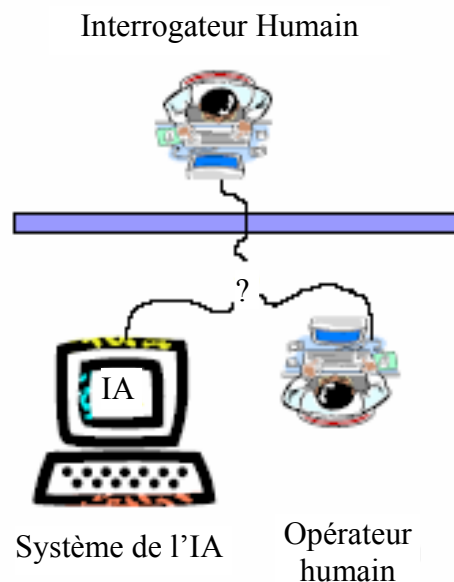


Figure 1- 1: Test de Turing.

5.3. Approche connexionniste (Numérique)

L'approche connexionniste est basée sur une modélisation des neurones qui constituent le cerveau humain. Elle utilise un ensemble de neurones artificiels, qui sont en réalité des unités de calcul, connectés entre eux par des liens dont chacun possède un poids. Ces poids seront ajustés pendant une phase nommée phase d'apprentissage qui sera réalisée par un ensemble d'exemples.

Les réseaux de neurones sont caractérisés par leur aptitude à apprendre à partir d'exemples et de pouvoir généraliser cet apprentissage sur d'autres exemples.

6. Système symbolique physique

Simon et Newell [Newell et al., 1981] définissent l'intelligence : « c'est une donnée générale des systèmes symboliques ».

Puis, ils ajoutent :

« Une structure de symboles est composée d'un nombre d'occurrences (ou signes) de symboles reliés d'une manière physique ».

« Un système de symboles physique est une machine qui produit dans le temps un assemblage évolutif de structures de symboles »

Enfin, ils définissent l'IA :

« Tout système de symboles physique possède les moyens nécessaires et

suffisantes pour une action intelligente de caractère général ».

Un Système symbolique physique est constitué d'un ensemble d'éléments appelés symboles ; Les structures symboliques (expression) sont constituées de symboles physiquement reliés ; Le système contient des procédures qui transforment (création, suppression, modification) les structures symboliques en d'autres structures.

Pour avoir un comportement intelligent, un système symbolique physique doit posséder les propriétés suivantes :

- Un symbole doit pouvoir désigner n'importe quelle expression ;
- Toute procédure que la machine est capable d'exécuter doit pouvoir être désignée par une expression ;
- Le nombre d'expressions n'est pas limité ;
- Les expressions peuvent générer ou modifier n'importe quelles expressions ;
- Elles sont stables.

7. Quelques techniques de l'IA

Les techniques de l'IA sont très nombreuses. Nous présentons ci-dessous, quelques unes qui ont fait ses preuves :

7.1. Système à base de règles

7.1.1. Définition

Plusieurs définitions de système à base de règles (système expert) existent, nous citons quelques unes :

« Un système expert est un système de calcul capable de représenter et de raisonner sur des domaines riches de connaissances, comme la médecine ou la géologie, avec comme objectif de résoudre des problèmes et de donner des conseils». [Jackson 1986]

« Un système expert est un programme possédant une grande masse de connaissances dans un domaine spécialisé, ces connaissances provenant généralement d'un expert humain chevronné et capable d'atteindre les performances de l'expert dans ce domaine ». [Bonnet et al., 1986]

Un système expert est un logiciel ou une partie d'un logiciel destiné à assister l'homme dans des domaines où est reconnue une expertise humaine. Dans les systèmes experts à base de règles cette connaissance est stockée dans la base de connaissances, qui est interprétée par le moteur d'inférence utilisant la base de faits pour stocker les données propres au problème à traiter et les déductions faites par le système. [Royer, 1990]

7.1.2. Composants d'un système expert

Un système expert est composé de trois composantes principales:

- *La base de règles*: elle contient l'ensemble de règles de productions et des connaissances heuristiques nécessaire à la résolution de problèmes. Elle ont la forme de (hypothèse, conclusion). Par exemple : si (p et q) alors r

- *La base de faits* : elle contient l'ensemble des faits relatifs au problème considéré. Ces connaissances (faits) sont acquises directement à partir d'un expert ou bien générées par le système lui-même. Elles représentent des connaissances spécifiques sur un domaine d'application particulier. Dans le cas de la logique propositionnelle, il s'agit en effet des affirmations concernant la valeur de vérité des propositions (p est vrai, q est faux).

L'ensemble de règles et de faits constitue la base de connaissances du système expert.

- *Un moteur d'inférence* : c'est un programme informatique qui est chargé d'exploiter la base de connaissances (faits et règles) pour effectuer un raisonnement et tirer des déductions. Il utilise des techniques de parcours par chaînage avant ou chaînage arrière selon le système. Il permet de générer de nouveaux faits à partir de règles et de faits déjà présents dans le système afin de résoudre un nouveau problème.

7.1.3. Quelques exemples de systèmes experts

Les systèmes experts sont actuellement très utilisés dans de nombreux domaines :

- En chimie : le système DENDRAL qui est le premier système expert développé en 1964 à l'université de Stanford. Son but est de donner la formule développée d'un composant chimique à partir de sa formule brute. Il a donné des résultats satisfaisants au cours de son utilisation.
- En médecine : avec le célèbre système expert MYCIN pour le diagnostic des maladies infectieuses. Ce système a été développé en 1974. Lui aussi, il a connu un grand succès.
- En conception : le système RITA développé par RAND corporation en 1976. C'est un système d'aide à la conception.
- En compréhension de la parole : avec le système HEARSAY-II.
- En géologie : avec le système PROSPECTOR développé en 1978.
- En maintenance : le système DART développé par IBM pour l'étude des pannes de matériels et de systèmes.

7.2. Algorithmes génétiques

7.2.1. Présentation

Inspirés des techniques d'évolution des gènes biologiques, Les algorithmes génétiques (appelés aussi algorithmes évolutionnaires) représentent une technique d'optimisation utilisée aussi pour la prédiction et la classification. Elle utilise des techniques imitées de la biologie, telles que la sélection naturelle, la recombinaison (croisement), la mutation et l'inversion, pour faire évoluer une population initiale. Cette technique est basée sur deux fonctions principales :

- *La fonction de codage* : Elle permet de représenter les données sous forme binaire. On trouve dans la pratique d'autres formes de représentation.
- *La fonction d'adaptation ou de fitness* : Elle permet de calculer l'adaptation d'une séquence de bits. Idéalement, elle vaut '1' si la séquence de bits est parfaitement adaptée et '0' si l'adaptation a échoué.

Cette technique est utilisée pour résoudre les problèmes où on ne peut pas les faire autrement.

7.2.2. Les opérations génétiques

Nous distinguons trois opérations principales utilisées par les algorithmes génétiques pour faire évoluer une population, à savoir :

- o *La mutation* : consiste à changer la valeur d'un bit au hasard d'une séquence elle-même sélectionnée au hasard. Si ce bit égal à '1' alors il sera converti à '0', dans le cas contraire, il sera converti à '1'.

▪ Exemple :

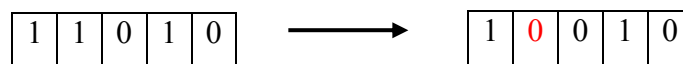


Figure 1- 2 : l'opération de mutation

- o *Le croisement* : consiste à permuter deux séquences à partir d'un bit choisi de manière aléatoire. Nous obtenons deux nouvelles séquences : la première est constituée de la première partie de la première séquence et de la deuxième partie de la deuxième séquence. L'autre séquence est constituée de la première partie de la deuxième séquence et de la deuxième partie de la première séquence.

▪ Exemple :

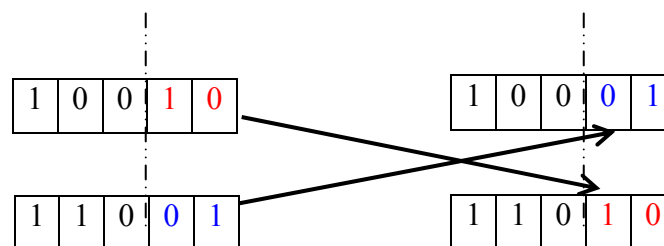


Figure 1- 3: l'opération de croisement

- o *L'inversion* : consiste à convertir deux caractères consécutifs d'une même séquence.

- Exemple :

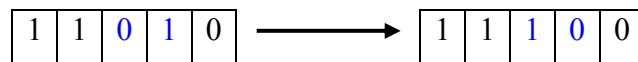


Figure 1- 4: l'opération d'inversion

7.2.3. Algorithme

L'algorithme génétique s'exécute en plusieurs étapes [Alliot et al., 2002] :

1. La génération aléatoire d'un nombre de séquences de bits pour créer la population initiale ;
2. Mesure de l'adaptation de chacune de ces séquences générées dans l'étape 1 ;
3. Reproduction de chaque séquence en fonction de son adaptation. Ces nouvelles séquences vont former la nouvelle population. Les séquences les mieux adaptées se reproduisent mieux que les séquences dont l'adaptation a échoué.
4. Des opérations de croisement et de mutation sont effectuées à des séquences tirées au hasard de la population.
5. retour à l'étape 2.

Ce processus se répète jusqu'à ce que nous obtenions une solution satisfaisante.

7.2.4. Inconvénients :

Cette technique présente quelques inconvénients, à savoir :

- elle est coûteuse en temps de calcul ;
- elle ne garantit pas l'optimalité des résultats ;
- elle demande une grande expérience.

7.3. Agents intelligents ou Knowbot

7.3.1. Qu'est ce qu'un agent ?

Le terme de Knowbot est un composé de Knowledge (connaissance) et Robot. Un agent est un système informatique, situé dans un environnement, et qui agit d'une façon autonome pour atteindre les objectifs pour lesquels il a été conçu [Wooldrige et al., 1995]. C'est une entité logicielle autonome capable d'agir sur elle-même et sur son environnement à la place d'un être humain et à son absence. Chaque agent dispose d'une représentation partielle de son environnement et peut communiquer avec d'autres agents. Il est guidé par un objectif individuel, et son comportement est étroitement relié par ses observations, ses compétences, et les interactions qu'il peut avoir avec d'autres agents de son environnement. Chaque agent possède un objectif personnel qui le distingue d'un programme informatique classique.

7.3.2. Caractéristiques d'un agent

Un agent est caractérisé par son :

- Intelligence : un agent peut raisonner et apprendre.
- Gérable : il prend ses instructions d'un opérateur humain ou d'un autre agent.
- Mobilité : un agent peut se déplacer sur un réseau informatique pour accomplir ses tâches. Cette caractéristique est très intéressante, supposant qu'il faut choisir une image parmi un très grand nombre d'images qui se trouvent dans un autre site; au lieu de toutes les télécharger pour enfin ne garder qu'une seule, ne vaut-il mieux pas d'envoyer un agent, qui connaît les préférences de cet utilisateur, sur la machine où les images sont stockées pour qu'il effectue le choix à sa place.
- Autonomie : un agent peut agir sans l'intervention d'un opérateur humain ou autre agent. Il contrôle son état et son comportement et préserve ses propres intérêts.
- Persistant : il sait ne rien faire sur de longues périodes.
- Fiable : il répond aux besoins de l'utilisateur.
- Prévoyant : il sait anticiper les besoins.
- Actif : il peut prendre des initiatives sans qu'il soit sollicité ou lancé par une autre entité.
- Communicant : il communique avec son environnement pour résoudre ses problèmes et atteindre ses objectifs.
- Adaptatif : il sait changer d'environnement et il peut s'adapter selon l'environnement.

7.3.3. Système multi-agents

Un système multi-agents est un système informatique composé d'un environnement, d'objets passifs et d'agents communiquant directement ou indirectement et pouvant opérer sur les objets de l'environnement. Le comportement d'un système multi-agents est le résultat des interactions et du comportement des agents qui le constituent.

7.3.4. Types d'agent

On distingue deux types d'agents :

- *Les agents réactifs*

Un SMA constitué d'agents réactifs possède généralement un grand nombre d'agents et présente un comportement global intelligent. Les agents réactifs sont considérés intelligents au niveau du groupe, du système. En conséquence, l'intelligence est distribuée entre beaucoup d'agents réactifs et le comportement intelligent devrait émerger de l'interaction entre ces agents réactifs et l'environnement.

Cependant, la convergence du comportement de l'ensemble des Agents vers un état stable n'est pas forcément assurée, et si un état stable est atteint, il n'est pas certain qu'il s'agisse de la solution optimale.

- **Les agents cognitifs**

Un SMA constitué d'agents cognitifs possède généralement peu d'agents. Ils réclament des ressources plus importantes que les systèmes d'agents réactifs.

La convergence du système vers un état décisionnel stable n'est pas non plus assurée par l'utilisation de ce type d'Agents, mais ils permettent de résoudre des problèmes plus complexes et nécessitant une plus grande abstraction.

Les agents cognitifs possèdent une représentation partielle de l'environnement, des buts explicites, ils sont capables de planifier leur comportement, mémoriser leurs actions passées, communiquer par envoi de messages, négocier, etc.

7.4. Apprentissage par renforcement

7.4.1. Définition

L'apprentissage par renforcement peut être défini comme le problème posé à un agent autonome (une personne, un animal, un robot, un système intelligent) évoluant au sein d'un environnement, qui recherche à travers des expériences itérées au sein de cet environnement un comportement optimale. Ce comportement est modélisé selon une stratégie qui est une fonction associant à l'état courant de l'agent et de l'environnement, une action à exécuter. La qualité de cette action est évaluée en fonction des récompenses reçues au cours du temps par l'agent suite à l'application de cette stratégies. [Alliot et al., 2002]

7.4.2. Composant d'un système d'apprentissage par renforcement

Un système d'apprentissage par renforcement requiert :

- Un module lui permettant de déterminer les comportements ou actions possibles en fonction de l'environnement (système à base de règles par exemple) ;
- Une fonction de gain associant un gain objectif (reward) à chaque situation réelle ;
- Une fonction d'évaluation associant à chaque situation possible de l'environnement une estimation anticipée de la récompense finale associée à un but fixé (c'est-à-dire, de la récompense globale correspondant à tous les gains de toutes les situations réelles rencontrées jusqu'à la réalisation du but) ;
- Eventuellement, un modèle de l'environnement permettant de simuler les actions pour en estimer les conséquences.

– Exemple

Considérons le problème de stationnement dans un parking très grand d'un supermarché, et qui se trouve à coté d'une voie à sens unique. Le but est de minimiser la distance entre la position de stationnement et l'entrée du magasin. Si on s'approche beaucoup du supermarché, on risque de ne pas trouver de place ce qui va nous imposer d'aller au parking souterrain. Jour après jour, le problème de choix de la position de stationnement sera dicté par un apprentissage par renforcement.

7.5. Arbres de décision

Les arbres de décision sont utilisés dans le cadre de la découverte de connaissances dirigée. Ce sont des outils très puissants principalement utilisés pour la classification, la description ou l'estimation.

7.5.1. Composants d'un arbre de décision

Un arbre de décision est composé :

- D'un nœud racine par lequel on commence le raisonnement,
- De questions,
- De réponses qui conditionnent la question suivante,
- De nœuds feuilles qui correspondent à un classement.

7.5.2. Fonctionnement

Le principe de fonctionnement est le suivant : pour expliquer une variable, le système recherche le critère le plus déterminant et découpe la population en sous populations possédant la même entité de ce critère. Chaque sous population est ensuite analysée comme la population initiale. Le modèle rendu est facile à comprendre et les règles trouvées sont très explicites.

7.6. Logique floue

7.6.1. Principe

La notion de logique floue vient de l'observation que les variables booléennes, qui ne peuvent prendre que deux valeurs (vrai ou faux) sont mal adaptées à la représentation de la plupart des phénomènes naturels. Alors que la logique classique considère qu'une proposition est soit vraie soit fausse, la logique floue distingue une infinité de valeurs de vérité (entre 0 et 1 compris). Elle permet de représenter des données imprécises et incertaines, et donc de rendre compte de concepts de la vie courante qui ne sont pas traités par les approches binaires classiques.

La logique floue permet de présenter les données imprécises. Supposons qu'on veut déterminer si 18°C est une température chaude ou froide. Dans la logique classique, on introduit une valeur dite de seuil. Au-dessus de cette valeur, la température sera chaude et au-dessous, elle sera froide. Une erreur de 1% sur la valeur du seuil peut provoquer une erreur de 100% sur la décision. La logique floue permet de caractériser une appartenance graduelle à un sous-ensemble ou sous-ensemble flou. Cela signifie que la théorie des ensembles flous permet d'éviter le seuillage et ainsi de combiner plusieurs critères de décision.

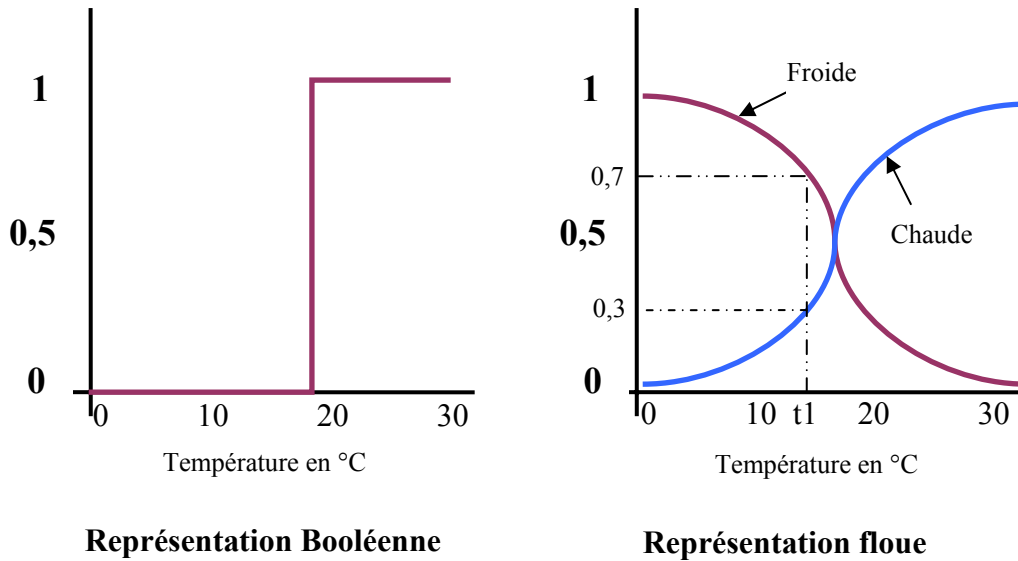


Figure 1- 5 : exemple comparatif entre logique booléenne/floue

Dans la logique floue, la température t_1 (Figure 2) est considérée comme une température froide à 30%, et chaude à 70%. Il n'y a pas un seuil qui sépare les températures froides des températures chaudes.

7.6.2. Fonctionnement

Le fonctionnement de la logique floue passe par trois phases principales :

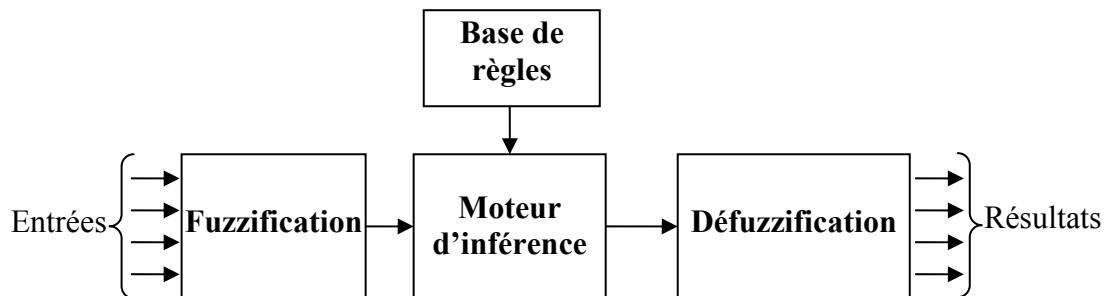


Figure 1- 6: processus d'un système flou.

- **Fuzzification**

Cette étape permet de déterminer le degré d'appartenance de chaque variable d'entrée à chaque état. Celui ci est déterminé à l'aide des fonctions d'appartenance définies dans le système. Ces fonctions d'appartenance, qui utilisent en général entre trois et sept états flous pour chaque variable, possèdent différentes formes : en triangles, en cloche ou encore en trapèze.

- **Base de règles floues**

Les degrés d'appartenance de chaque variable à chaque état permettent d'appliquer les règles floues qui ont été préalablement définies, elles sont de la forme (Si alors). Le degré d'appartenance des variables de sortie à chaque état est ainsi obtenu. Les règles peuvent mettre en jeu plusieurs variables dans leurs

conditions et leurs conclusions et la détermination du degré d'appartenance de chacune des conditions des règles se fait par inférence.

- **Défuzzification**

Elle transformation des résultats flous en sorties précises. Cette étape s'effectue toujours à l'aide des fonctions d'appartenance. A partir des degrés d'appartenance, on obtient autant de valeurs qu'il y a d'états. Pour déterminer la valeur précise à utiliser, on peut soit conserver le maximum, soit calculer la moyenne pondérée, soit déterminer le centre de gravité des valeurs obtenues.

7.7. Data mining

7.7.1. Définition

L'Extraction de Connaissances à partir de Données (ECD), plus communément appelée Data Mining, est un terme Français utilisé pour décrire le processus d'acquisition de connaissances dans les bases de données (Knowledge Discovery Databases, en anglais). Une confusion existe entre l'ECD et le Data Mining. Ce dernier ne serait en effet qu'un maillon du processus d'ECD selon les spécialistes du domaine [Rakoto 2004].

« Le data mining est l'ensemble du processus d'extraction de connaissances à partir de données contenues dans une base de données » [Lefébure et al., 2001].

Le terme de Data Mining est souvent employé pour désigner l'ensemble d'outils et de méthodes d'aide à la décision, permettant à l'utilisateur d'accéder aux données, de les analyser et de détecter dans ces données des règles et des informations cachées dans des gigantesques bases de données.

On pourrait définir le Data Mining comme une démarche ayant pour objet de découvrir des relations et des faits, à la fois nouveaux et significatifs, sur de grands ensembles de données. Il permet, grâce à un certain nombre de techniques spécifiques, de faire apparaître de nouvelles connaissances.

7.7.2. Les tâches du Data Mining

Le data mining est un outil qui permet :

- Classification : c'est l'affectation des éléments à des classes prédéfinies selon leurs caractéristiques.
- Segmentation (ou clusterisation) : la segmentation consiste à fragmenter un ensemble hétérogène en sous-ensembles homogènes. Ces sous-ensembles ne sont pas prédéfinis contrairement à la classification.
- Optimisation : Le but de l'optimisation est de maximiser ou minimiser une certaine fonction (fonction d'évaluation).

7.8. Réseaux bayésiens

7.8.1. Principe

Les réseaux Bayésiens ont pour objectif la découverte des relations. Ils permettent en effet la compréhension de certaines relations. Ils sont basés sur la théorie des graphes.

7.8.2. Fonctionnement

Un réseau Bayésien est un graphe orienté dans lequel les nœuds représentent les variables et les arêtes symbolisent les dépendances entre les variables. Il mesure la probabilité d'apparition d'un événement connaissant le résultat observé sur d'autres variables.

7.8.3. Mise en place

La mise en place d'un réseau Bayésien passe par les étapes suivantes :

- Préparation des variables,
- Sélection des variables,
- Identification des dépendances,
- Matrice des probabilités.

7.9. Raisonnement à partir de cas

7.9.1. Principe

Le raisonnement à partir de cas (RÀPC) est basé sur la réutilisation des expériences passées qui sont capitalisées dans une mémoire appelée base de cas pour résoudre de nouveaux problèmes tout en sélectionnant un cas similaire à la situation en cours, et en adaptant par la suite l'ancienne solution pour laquelle correspond au nouveau problème.

La base de cas est constituée par les anciens cas qui sont déjà été résolus. Pour chacun de ces cas, on garde une description du problème et de sa solution. Chaque description est une suite d'attributs valeurs.

7.9.2. Processus RÀPC

Le processus de RÀPC comprend cinq étapes séquentielles qui sont : Elaboration, Remémoration, Adaptation, Révision et Mémorisation.

A l'arriver d'un nouveau problème, et après son élaboration, le système de RÀPC va chercher dans sa base de cas, un cas dont la partie problème est la plus similaire au problème en cours (remémoration). Ensuite, il va réutiliser la partie solution du cas sélectionné dans la phase précédente (remémoration) après qu'elle soit adaptée à la nouvelle situation (adaptation). Cette dernière va être révisée par l'utilisateur de système RÀPC avec une éventuelle modification (révision) et finalement, cette solution révisée va être capitalisée avec le problème en cours dans la base de cas si elle est jugée pertinente (mémorisation).

Le chapitre suivant va être consacré pour décrire cette technique en détail.

7.10. Réseaux de neurones

7.10.1. Présentation

Le fonctionnement d'un réseau de neurones est inspiré de celui du cerveau humain. Chaque neurone reçoit des impulsions, il les traite, et le résultat obtenu sera transmis aux autres neurones connectés à sa sortie.

Le *neurone formel* (artificiel) est une unité de calcul qui fait la somme pondérée par le poids de l'arc de chaque entrée transmise par le reste du réseau.

Un réseau de neurones est un ensemble de neurones connectés entre eux, et qui échange des données et réalisent des calculs. Chaque réseau possède une couche d'entrée où il reçoit les données de son environnement, et une couche de sortie pour fournir le résultat.

7.10.2. Processus du raisonnement

Avant d'être exploités, les réseaux de neurones doivent tout d'abord apprendre. Cet apprentissage sera réalisé par un ensemble d'exemples qui doivent être bien choisis pour avoir de bons résultats dans la phase d'exploitation du réseau. Trois types d'apprentissage sont distingués : Apprentissage supervisé, Apprentissage semi-supervisé, Apprentissage non supervisé.

Après cette phase, le réseau de neurones peut être exploiter avec des exemples même qui n'appartient pas à l'ensemble d'exemples utilisés dans la phase précédente.

Cette technique sera présentée en détail dans le chapitre 3.

8. Quelques difficultés rencontrées en IA (limites)

La majorité des techniques de l'IA doivent faire face à deux problèmes majeurs qui sont intimement associés à l'IA, à savoir :

8.1. Complexité

La théorie de la *complexité algorithmique* s'intéresse à l'estimation de l'efficacité des algorithmes. Elle est souvent attachée à la question : entre différents algorithmes réalisant une même tâche, quel est le plus rapide et le moins consommable en mémoire vive ?

Elle repose sur la définition de classes de complexité qui permettent de classer les problèmes en fonction de la complexité des algorithmes qui existent pour les résoudre. Plusieurs classes de complexité sont définies : Classe L, Classe NL, Classe P, Classe NP, Classe Co-NP (Complémentaire de NP), Classe PSPACE, Classe NSPACE ou NPSPACE, Classe LOGSPACE, Classe EXPTIME...etc.

La plupart des programmes qui implémentent des techniques de l'IA ont une complexité très élevée par rapport à d'autres programmes qui résolvent les mêmes problèmes sans utiliser des techniques de l'IA. Néanmoins, la majorité des problèmes traités par les techniques de l'IA ne peuvent pas être résolus autrement.

8.2. Explosion combinatoire

Explosion combinatoire se traduit par le fait qu'un petit changement du nombre de données utilisées dans un problème par ailleurs trivial peut suffire à rendre sa solution très difficile, voire impossible d'être réalisée.

La plupart des programmes en IA doivent affronter le problème de l'explosion combinatoire qui provoque l'explosion du nombre d'états à explorer par un programme. Ce type de problème peut se poser comme dans le cas d'inférence dans les systèmes experts, le parcours d'un arbre de décision...etc.

Une des manières de réduire la combinatoire consiste à réduire le nombre des appariements autorisés en utilisant les divers attributs qui caractérisent les primitives. Une autre consiste, dans le cas où des calculs identiques et longs ont une grande probabilité d'être répétés, de mettre en mémoire leurs résultats pour éviter de les

recalculer.

9. Domaines concernés par l'IA

Les applications de l'IA sont très nombreuses, et touche différents domaines, parmi ces applications, nous citons :

9.1. Compréhension ou traitement du langage naturel (NLP ou TLN)

Les recherches en TLN s'intéressent à la compréhension des langages naturels, à la traduction automatique d'un langage naturel à un autre par l'utilisation des techniques de l'IA, la reconnaissance de la parole, la représentation des connaissances.

9.2. Enseignement à distance (E-learning)

Les dernières années ont connu un fort essor de la FOAD (Formation Ouverte À Distance) dû essentiellement au développement des TICE (Technologies de l'Information et de la Communication dans l'Enseignement). La FOAD est devenue plus privilégiée par les apprenants qui préfèrent des modalités d'apprentissage plus souples, adaptées à leurs besoins avec une sensation de liberté et d'autonomie.

C'est un domaine où l'IA a commencé de s'imposer en adaptant et gérant les ressources utilisées dans ce type de formation.

9.3. Démonstration et découverte automatique de théorèmes

Ce domaine est bien formalisé, bien maîtrisé par les premiers chercheurs en IA. C'est un domaine typique de l'IA et des résultats sont obtenus dès les début de recherches. Les premiers programmes sont apparus en 1956.

Des méthodes sont utilisées pour démontrer des théorèmes telle que Le Principe de Résolution. Il a donné naissance au langage PROLOG. Pratiquement, de nombreuses heuristiques doivent être utilisées pour limiter l'explosion combinatoire. Les méthodes naturelles sont plus proches du raisonnement humain et rendent plus faciles la donnée de l'expertise du mathématicien. Les performances actuelles sont supérieures à celles d'un individu moyen, inférieures à celles d'un bon mathématicien.

Ce domaine dépasse le cadre des mathématiques et est utilisé dans de nombreux domaines, par exemple en Résolution de problèmes et en Vérification de programmes.

9.4. Robotique

Les applications de la robotique sont particulièrement nombreux : travaux dangereux, réparations dans les centrales nucléaires ou dans l'espace...etc.

9.5. Programmation Automatique et vérification de programmes

Les techniques de L'IA ont permis de créer des programmes qui écrivent des programmes résolvant des problèmes décrits en langage mathématique ou en langage naturel. Ces programmes sont souvent moins bons que ceux écrit par des programmeurs humains, mais ils peuvent être meilleurs. Ce domaine a été pendant un temps concurrencé par les systèmes experts, ils sont maintenant complémentaires.

9.6. Autres domaines

- Médecine ;
- Programmation des jeux ;

- Résolution de problèmes ;
- Traitement d'images ;
- Reconnaissance de formes ;
- Vision artificielle et perception ;
- Interaction Homme/Machine ;

10. Principaux langages de l'intelligence artificielle

Beaucoup de langages permettent la mise en place des techniques de l'IA, parmi ces langages, nous citons :

- OCaml : langage fonctionnel ;
- Lisp (1960, J. McCarthy) ;
- Prolog (1973, A. Colmerauer), Prolog avec contraintes ;
- SmallTalk (1972, A. Kay) ;

11. Conclusion

Dans ce chapitre, nous avons présenté le domaine de l'intelligence artificielle avec ses approches, ses techniques les plus utilisées, les domaines où ces techniques sont très appliquées et les difficultés rencontrées pendant la mise en œuvre de l'une de ces techniques.

Dans le chapitre suivant, nous allons détailler la technique du raisonnement à partir de cas puisque notre travail s'inscrit dans le cadre de ce paradigme de l'intelligence artificielle.

Chapitre II

Raisonnement à partir de cas

1. Introduction

Pour résoudre les problèmes qu'on rencontre dans la vie quotidienne, nous faisons généralement appel à nos expériences. Nous nous remémorons les situations semblables déjà rencontrées. Puis nous les comparons à la situation actuelle pour construire une nouvelle solution qui, à son tour, s'ajoutera à nos expériences.

Les expériences constituent un capital important pour les humains, et si cette ressource primordiale sera bien capitalisée, nous pouvons gagner beaucoup de temps et résoudre mieux nos problèmes, et nous pouvons aussi la faire circuler entre les gens qui ont besoin d'elle.

Le raisonnement à partir de cas (RàPC) part de ce principe. Il a pour but de capitaliser les anciennes expériences en vue de les adapter et réutilisées pour la résolution de futurs problèmes rencontrés.

Ce chapitre sera organisé comme suite : nous définirons le paradigme du RàPC dans la section suivante. Ensuite, nous citons quelques conditions pour que le système RàPC soit efficace (section 3). Dans la section 4, nous rappellerons les différents types de connaissances utilisées par un système RàPC. Après (section 5), nous citerons les différentes étapes du processus du raisonnement. Les sections 6 et 7 seront consacrées pour présenter respectivement les modèles de représentation de cas et les modèles d'organisation de la base de cas. Dans les deux dernières sections (8 et 9), nous évoquerons respectivement quelques projets et quelques rencontres sur le RàPC.

2. Définition

Le raisonnement à partir de cas (RàPC), en anglais Case Based Reasoning (CBR), est inspiré du raisonnement par analogie, il trouve ses origines dans les travaux de Roger Schank sur la mémoire dynamique [Schank 1982]. Le RàPC repose sur l'hypothèse que « des problèmes similaires ont des solutions similaires ». Le RàPC est un cas particulier de raisonnement par analogie.

Le RàPC est une approche de résolution de problèmes basée sur la réutilisation d'expériences passées appelées "cas sources" pour la résolution de nouveaux problèmes appelés "cas cibles".

Un cas est décomposé en deux parties, la première décrit le problème et la seconde sa solution. Chaque partie est représentée par un groupe de descripteurs où chaque descripteur est formé d'un couple (attribut, valeur). L'ensemble des cas constitue la base de cas.

3. Quand utilise t'on un système de RàPC ?

Le RàPC est généralement bien adapté pour les applications ayant les caractéristiques suivantes :

- les tâches sont bien accomplies par des humains expérimentés dans leur domaine et ces expériences sont disponibles dans une base de données, dans des documents ou chez un expert humain ;
- une analyse détaillée du domaine n'est pas nécessaire pour obtenir des solutions satisfaisantes et la tâche n'exige pas une solution optimale (satisficing solution) ;

- un modèle du domaine peut ne pas être élaboré parce que le domaine est mal formalisé (peu de documentation, expert non disponible) ou parce qu'il n'existe pas de principes généraux qui sont éprouvés,
- les situations sont répétitives et les solutions sont réutilisables. Ces situations dites monotones, sont telles que les petites différences dans le problème entraînent de petites différences dans la solution.

4. Typologie de connaissances

Nous distinguons quatre types de connaissances indispensables pour la mise en place d'un système de RàPC, à savoir :

4.1. Les cas

Les cas constituent le noyau d'un système de raisonnement à partir de cas. Ils présentent les expériences vécues et capitalisées pour être réutilisées dans les futures résolutions. L'utilisateur d'un système de RàPC ne s'intéresse qu'à ce type de connaissances. Les autres types de connaissances sont généralement invisibles pour lui puisqu'elles sont utilisées que par le système de RàPC.

4.2. Les connaissances du domaine (Vocabulaire d'indexation)

Ensemble d'attributs qui caractérisent la description de problèmes et de solutions du domaine. Ce type de connaissances joue un rôle important dans la phase de remémoration.

4.3. Les connaissances de similarité

Les connaissances de similarité prennent plusieurs formes telles que chemins de similarité, distance euclidienne ou distance de Hamming entre les cas, des index, des questions dans un arbre de décision...

Ces connaissances sont utilisées pendant la phase de remémoration pour la sélection de cas similaires à un problème donné.

La technique de K plus proches voisin (KPP) est une façon pour mesurer la similarité entre les cas. Elle utilise un calcul de distances pour affecter un élément à une classe parmi les classes existantes.

4.4. Les connaissances d'adaptation

Ces connaissances sont généralement de nature implicite et leur formalisation sous forme explicite est une tâche très difficile.

Elles peuvent prendre plusieurs formes, telles que : les règles d'adaptation, les dépendances entre les attributs du problème et ceux de sa solution, les cas d'adaptation.... Elles sont générées à partir de la base de cas ou formalisées par le concepteur du système.

Elles sont utilisées pour modifier la solution source pour quelle correspond au nouveau problème.

5. Processus de raisonnement à partir de cas

Le RàPC consiste à sélectionner un ou plusieurs cas similaires au nouveau cas ; d'adapter, de réutiliser et de réviser la ou les solutions relatives au cas sélectionnés; et enfin si le nouveau cas est jugé pertinent, il sera capitalisé dans la base de cas.

Le processus de RàPC comporte 5 étapes séquentielles qui sont : (1) l'élaboration ; (2) la remémoration ; (3) l'adaptation; (4) la révision ; (5) l'apprentissage (Figure 2-1) :

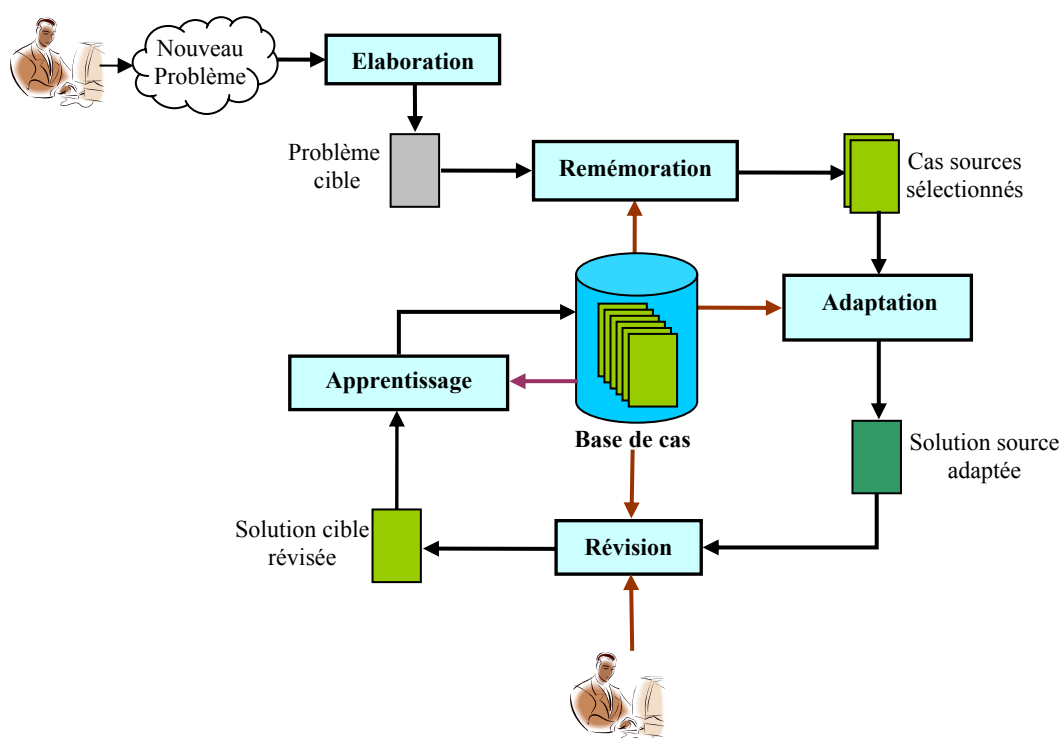


Figure 2- 1: Cycle RàPC

Nous décrivons ci-dessous chacune de ces étapes :

5.1. Élaboration

L'élaboration des cas consiste à formuler le nouveau problème à résoudre appelé problème cible pour qu'il décrive au maximum la situation courante et pour qu'il soit cohérent par rapport à la base de cas. C'est une modélisation du nouveau problème en réponse à ses spécifications et en conformité avec la représentation des cas de la base. Lors de cette étape, le système de RàPC demande à l'utilisateur ou à son environnement (bases de données, entrepôt de données...) de lui fournir l'ensemble des informations nécessaires à la poursuite du raisonnement.

5.2. Remémoration

La remémoration consiste à rechercher dans la base de cas un cas source dont la partie problème est jugée la plus similaire au problème cible afin de réutiliser sa partie solution pour résoudre le nouveau problème. La sélection des cas similaires s'appuie sur une mesure de similarité. Certains systèmes conservent plusieurs cas remémorés pour les combiner par la suite, d'autres n'utilisent qu'un seul cas pour la résolution du problème. Le choix d'un cas source parmi les cas remémorés peut être effectué par le

système RàPC ou bien par l'utilisateur du système.

5.3. Adaptation

L'adaptation est l'une des étapes les plus difficiles du RàPC [Cordier et al., 2006]. Elle permet d'obtenir une solution au problème cible à partir de la solution du cas source sélectionné au cours de la phase de remémoration. Cette dernière est tout d'abord recopiée puis éventuellement adaptée afin de satisfaire les contraintes du problème posé et de tenir compte des différences entre les problèmes source et cible. L'adaptation s'appuie sur des connaissances d'adaptation qui peuvent prendre différentes formes selon les systèmes. Ces connaissances sont généralement de nature implicite et leur explicitation sous forme de règles d'adaptation reste une tâche difficile.

Dans cette étape, la majorité des systèmes RàPC sollicitent l'aide des utilisateurs pour participer à la création d'une nouvelle solution à partir de la solution remémorée. Ceci est dû à la difficulté de l'implémentation de cette phase [Lieber et al., 2004].

Le but de l'adaptation est donc de proposer une solution au problème cible en modifiant la solution de problème source. Donc, on assiste à la présence d'un système de raisonnement par analogie:

« Sachant que la solution du problème cible est à problème cible ce que la solution du problème source est à problème source, connaissant problème source, solution du problème source et problème cible, que vaut la solution du problème cible ? » (Figure 2-2).

Le carré d'analogie peut être lu autrement :

« Sachant que la solution du problème cible est à la solution du problème source ce que problème cible est à problème source, connaissant problème source, solution du problème source et problème cible, que vaut la solution du problème cible ? » (Adaptée de [Carbonell 1983 ; Carbonell 1986]).

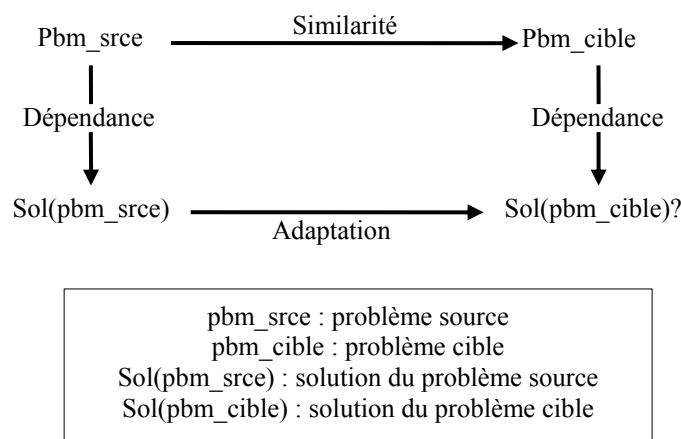


Figure 2- 2 : Schéma d'adaptation.

Il existe deux dimensions d'adaptation:

5.3.1. La dimension produit (transformationnelle, structurelle)

La solution trouvée n'est pas une solution directe pour le nouveau problème, mais il existe une connaissance sous forme des opérateurs transformationnels qui sont appliqués à la solution ancienne. On distingue deux sous classes dans

l'adaptation transformationnelle:

- *Par valeur*: elle consiste à adapter (changer) les valeurs des attributs de la solution du cas source.
- *Par structure*: elle consiste à adapter la structure de la solution du cas source (nombre d'attributs).

5.3.2. La dimension processus (dérivationnelle, générative)

Elle consulte les informations existant dans le cas remémoré. Ces informations contiennent la méthode utilisée pendant la résolution de l'ancien problème. Il s'agit donc, d'exploiter la démarche de résolution (le processus de raisonnement) utilisée dans la résolution de l'ancien problème pour créer une nouvelle solution afin de satisfaire le nouveau problème. Dans cette classe d'adaptation, on garde pour chaque cas passé une trace des étapes qui ont permis de générer sa solution. Cette traçabilité est souvent utilisée en logique de conception (Design rationale).

Les deux dimensions dérivationnelle et transformationnelle ont été introduites dans [Carbonell, 86] comme deux processus différents d'adaptation.

5.4. Révision

Pendant cette phase, l'utilisateur du système RàPC a le choix d'accepter ou de refuser la solution adaptée proposée par le système RàPC soit parce qu'elle est jugée non pertinente ou bien après qu'elle soit testée, elle ne va pas fournir une solution satisfaisante aux besoins d'utilisateur. On doit identifier les causes éventuelles des échecs du système et proposer des adaptations additionnelles pour aboutir à des solutions pertinentes. Si on exploite cette étape convenablement, on peut tirer beaucoup de connaissances en vue d'augmenter l'efficacité du système RàPC et minimiser ses échecs.

Dans cette phase, on pourra aussi évaluer l'adaptation, si la plupart des solutions adaptées sont remises en cause, il faut revoir les étapes précédentes.

L'étape de révision permet également d'évaluer l'utilité et la pertinence du cas nouvellement résolu et d'élaborer une stratégie de rétention ou d'oubli des cas selon leur contribution à la compétence du système [Lieber et al., 2003].

L'utilisateur est l'auteur principal de cette phase, ce qui explique la présence d'un opérateur humain qui intervient dans le schéma (Figure 2-1). Si ces interactions seront bien exploitées, de nouvelles connaissances utiles vont apparaître.

5.5. Apprentissage

Au cours de cette phase, Appelée aussi phase de mémorisation, on cherche à faire évoluer la base de cas en prenant en compte le problème qui vient d'être résolu et qui pourra ainsi être utilisé pour les futurs raisonnements. Il s'agit de mesurer l'utilité du nouveau cas en vue de le capitaliser. Cette phase implique une mise à jour des index permettant de remémorer des cas et prévoit un processus de maintenance pour réordonner la base de cas (ajout, substitution, rejet).

La mémorisation ne se limite pas aux cas, d'autres types de connaissances telles que les connaissances de similarité ou d'adaptation peuvent être acquises durant cette étape.

6. Modèles de représentation de cas

La représentation des cas prend une place primordiale dans la réalisation d'un système RàPC. En effet cette représentation va influencer directement sur l'efficacité et la rapidité de la recherche des cas dans la base. Il est donc nécessaire de choisir les informations à stocker dans chaque cas et de trouver sous quelle forme.

Il existe plusieurs modèles pour la représentation des cas dans les systèmes de raisonnement à partir de cas. Ces modèles sont regroupés en trois grandes familles : structurelle, conversationnelle et textuelle [Lamontagne 2001, Lamontagne et al., 2002]. Nous présentons ces trois modèles ci-dessous :

6.1. Modèle structurel

Dans ce modèle, les caractéristiques (attributs) représentant les cas sont déterminées à l'avance dès la conception du système RàPC. Ainsi, le concepteur du système prévoit un modèle de données du domaine d'application pour structurer et représenter les cas sous la forme (attribut, valeur).

Ce modèle est le plus utilisé vu la facilité de sa mise en œuvre. Tous les travaux sur l'adaptation de cas sont menés dans le cadre du modèle structurel [Lamontagne 2004].

L'exemple suivant (Figure 3) représente un cas décrit par le modèle structurel :

N° de cas	003
Marque	Renault
Type	Clio
Puissance	9
Energie	Diesel
Couleur	Grise
Année	2006

Tableau 2- 1: Un cas voiture représenté par le modèle structurel.

6.2. Modèle conversationnel

Dans le modèle structurel, les cas doivent être décrits ou structurés avant d'entamer la phase de remémoration. Cette exigence présuppose une expertise du domaine d'application permettant de bien caractériser une situation à l'aide de valeurs numériques ou symboliques et de sélectionner les principaux facteurs pouvant influencer la résolution de son problème [Lamontagne 2004].

Cependant, il est parfois très difficile, voir impossible de déterminer les caractéristiques d'un problème dès le début du raisonnement. C'est pour cette raison que le modèle conversationnel a été proposé par « Inference Corporation » pour remédier ces difficultés.

Ce modèle de représentation répond bien aux applications commerciales (e-commerce) qui sont basées sur des systèmes de RàPC et qui nécessitent une assistance des clients surtout ceux qui sont nouveaux.

Comme son nom l'indique, le modèle RàPC conversationnel est basé sur l'interaction entre l'utilisateur et le système (d'où la notion de "conversation") pour définir progressivement le problème à résoudre et pour sélectionner les solutions les plus appropriées [Aha et al., 2001].

La construction d'un cas conversationnel se fait en trois étapes :

- un problème : une brève description textuelle, habituellement de quelques lignes, de la nature du problème exprimée.
- une série de questions et de réponses : des index, exprimés sous forme de questions, permettant d'obtenir plus d'information sur la description du problème. Chaque question a un poids représentant son importance par rapport au cas.
- une action : une description textuelle de la solution à mettre en oeuvre pour ce problème. Cette description n'est pas structurée.

Ce modèle de représentation de cas ne permet pas de faire l'adaptation. Ceci est dû à la représentation non structurée des solutions.

La conversation entre le système RàPC et son utilisateur se fait comme suit :

- Tout d'abord, l'utilisateur fournit au système une description textuelle du problème à résoudre. A partir de cette description, le système calcule la similarité entre cette dernière et les descriptions des problèmes de la base de cas. Ensuite, le système va proposer à l'utilisateur une série de questions.
- L'utilisateur choisit les questions auxquelles il veut répondre. Après chaque réponse fournie par l'utilisateur, le système recalcule la similarité de chacun des cas avec ce nouveau cas.
- Lorsqu'un des cas atteint un niveau de similarité suffisamment élevé (c'est à dire qu'il franchit un seuil), il sera proposé par le système comme solution à la situation courante. Si aucun cas n'atteint un degré de similarité suffisant et que le système n'a plus de questions à poser à l'utilisateur, le problème est stocké comme étant non résolu.

6.3. Modèle textuel

Ce modèle permet de représenter et d'exploiter les cas qui sont décrits dans des documents textuels. Ces documents regroupent des cas textuels non structurés ou semi structurés. Ils sont non structurés si leur description est complètement en texte libre. Ils sont semi structurés lorsque le texte est découpé en plusieurs portions étiquetées par des descripteurs tels que "problème", "solution", etc.

Cette représentation est importante puisque à l'aide d'elle, on peut décrire des situations (des cas) qui ne peuvent être facilement codifiées selon un schéma de représentation de connaissance. Elle est la plus récente parmi les modèles de représentations cités ci-dessus.

Nous identifions deux axes importants dans les différents travaux en RàPC textuel:

- *structuration de cas textuels* : on représente les textes selon un nombre limité de traits basés sur des caractéristiques du domaine (concepts, catégories, sujets, mots-clés, etc.). Pour cet axe de recherche, on vise à structurer le mieux possible les cas textuels afin de tirer profit de techniques

développées pour les systèmes RàPC structurel. Les efforts sont déployés pour enrichir l'indexation des textes à l'aide de traitements relativement élaborés. Cette approche est intéressante pour les applications dont le domaine est restreint.

- *extension du modèle de recherche d'information* : dans cet axe de recherche, on s'intéresse à l'élaboration des mécanismes de recherche plus sophistiqués tout en gardant le processus d'indexation le plus simple possible. Dans ce cadre, le choix des traits de cas est déterminé à partir de la fréquence de mots-clés. Les particularités de l'application se reflètent au niveau de la recherche, soit par la définition de mesures de similarité sémantique ou par des extensions au modèle vectoriel de recherche d'information [Salton et al., 1984]. Cette approche semble plutôt valide pour les applications génériques qui veulent conserver une indépendance par rapport au domaine d'application.

Les travaux actuels se concentrent principalement sur la phase de remémoration sur la base de cas. Cependant, la phase de l'adaptation de solutions textuelles est complètement ignorée par ces travaux du fait de la non structuration des solutions.

7. Modèles d'organisation de la base de cas

La base de cas doit être organisée et indexée selon un modèle. Ce modèle doit respecter certaines contraintes :

- L'ajout d'un cas ne doit pas empêcher l'accès aux anciens cas.
- La complexité pour la recherche d'un cas similaire doit être constante au fur et à mesure que la base de cas se grandisse.
- Les opérations telles que l'ajout, la suppression ou la substitution des cas doivent être faciles et rapides.

Il existe plusieurs modèles pour organiser la base de cas, parmi ces modèles, on trouve :

7.1. Modèle simple

Nous distinguons plusieurs représentations de la base de cas qui sont plus ou moins simples par rapport à d'autres qui sont plus complexes. Parmi ces représentations simples, nous citons :

- *La représentation linéaire des cas* : Cette représentation est toujours combinée avec d'autres modèles plus complexes pour représenter des sous ensembles de cas. Parmi les représentations linéaires, on trouve les tableaux (Array), les listes chaînées, les piles...etc.
- *La représentation sous forme d'arbre de décision* : chaque nœud correspond à une question sur l'un des index, et les fils correspondent aux différentes réponses. Pour que cette représentation soit efficace, les questions, dans l'arbre, doivent être posées dans le bon ordre et l'arbre doit être le moins profond possible.
- *La représentation sous forme d'une hiérarchie de prototypes* : dans cette représentation, Tous les cas dont les attributs vérifiant des conditions données sont associés à un prototype. La base de cas sera donc représentée

sous forme d'une hiérarchie d'héritage entre les prototypes. Cette représentation peut être combinée avec d'autres représentations telles que les arbres de décision pour obtenir une représentation intéressante.

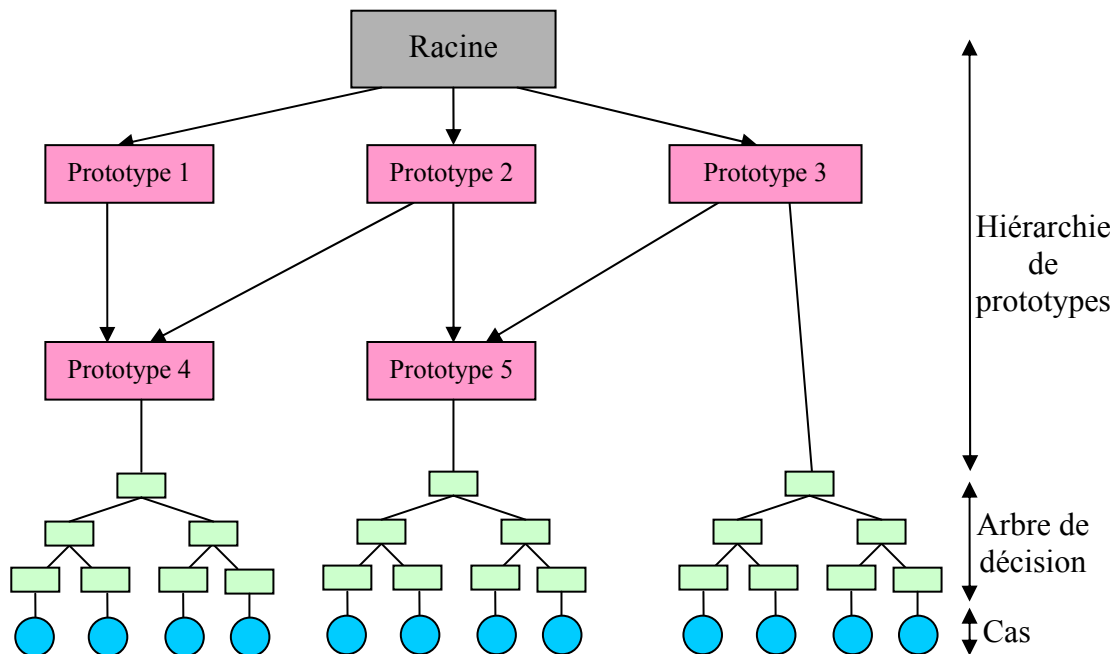


Figure 2- 3: représentation hiérarchique de prototypes combinée avec les arbres de décision

7.2. Modèle à mémoire dynamique

Ce modèle a été introduit par Robert Schank et Janet Kolodner. Dans ce modèle, les cas sont stockés dans une structure hiérarchique appelée épisode généralisé. Tous les cas ayant des propriétés similaires sont regroupés dans un épisode généralisé. Ces derniers contiennent trois types d'objets :

- *Les normes* : Les caractéristiques communes à chacun des cas indexés sous l'épisode généralisé.
- *Les index* : Les éléments discriminant les cas contenus dans l'épisode généralisé. Un index possède deux champs : son nom et sa valeur. Il peut pointer vers un autre épisode ou simplement vers un cas.
- *Les cas* : Les anciens problèmes avec leur solution. Ils peuvent être atteints par l'intermédiaire des index.

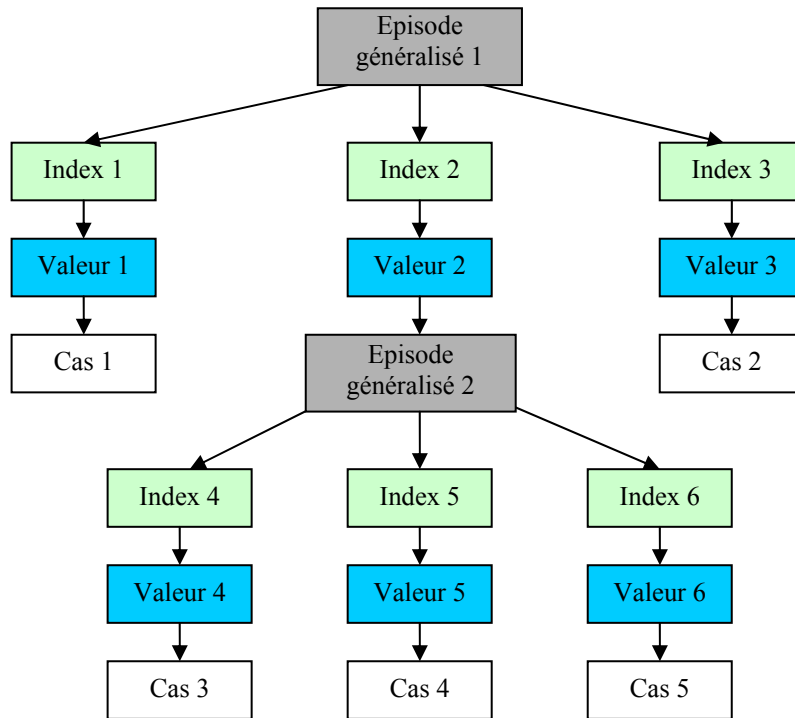


Figure 2- 4 : Représentation par le modèle à mémoire dynamique.

Dans ce modèle, les cas sont en effet, stockés dans les feuilles de cette structure.

7.3. Modèle à base de catégories

Dans ce modèle, la base de cas est un réseau de catégories et de cas. Les index sont des liens qui peuvent être de trois sortes :

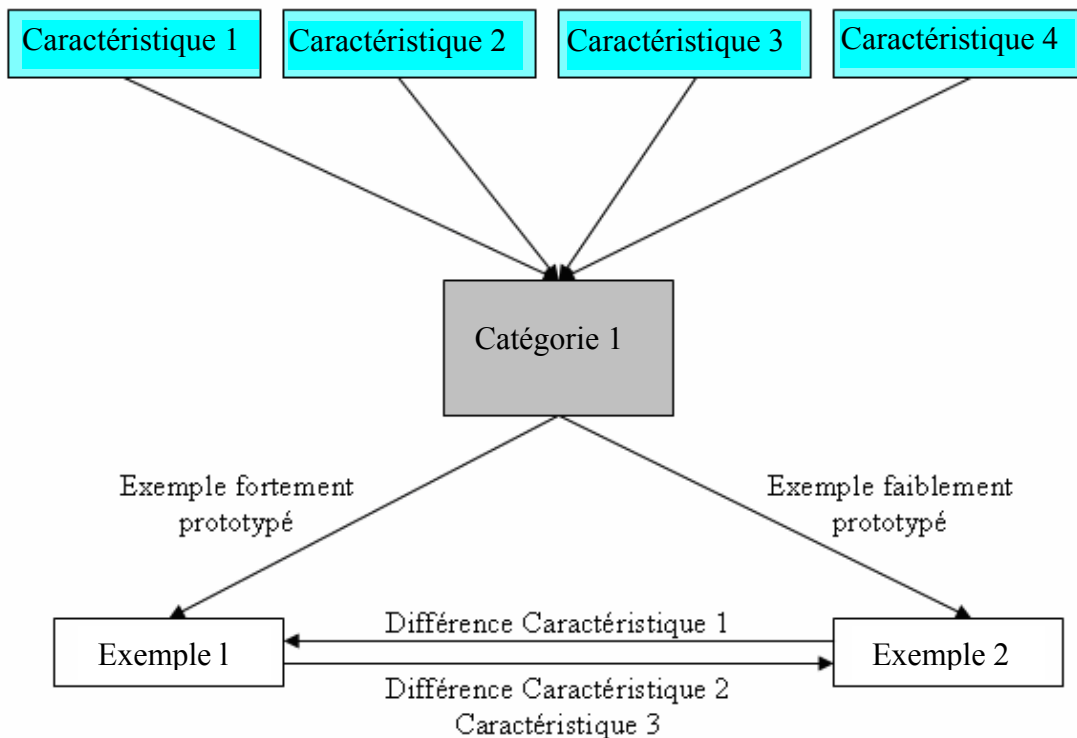


Figure 2- 5: Représentation à base de catégories

- *De rappel* : reliant une caractéristique à une catégorie ou un cas.
- *D'exemple* : reliant une catégorie aux cas auxquels elle est associée.
- *De différence* : reliant deux cas ne différant que d'un nombre restreint de caractéristiques.

8. Quelques projets en RàPC

De nombreux projets utilisent la technique de RàPC pour la résolution des problèmes. Parmi ces projets, on cite :

- **Le projet CYRUS**

Développé par Janet Kolodner [Kolodner 1982], CYRUS est historiquement le premier projet en RàPC basé sur le modèle de mémoire dynamique de Shank. C'est un système de questions/réponses qui retrace des événements impliquant Cyrus Vance dans son rôle de secrétaire d'état. Il implante le modèle de Schank, c'est à dire une structure de mémoire dynamique pour la recherche d'informations. Ce modèle de la mémoire de cas développé pour ce système a été la base de plusieurs autres systèmes de RàPC.

- **Le projet Kasimir**

Le projet Kasimir est une plate-forme logicielle regroupant l'équipe LORPAILLEUR, le CAV (Centre Alexis Vautrin de Nancy, spécialisé en cancérologie), ONCOLOR (réseau de santé en cancérologie de Lorraine) et le laboratoire d'ergonomie du CNAM de Paris. Il a pour objectif la gestion des connaissances contenues dans les référentiels qui sont des protocoles de décision en cancérologie à l'aide d'un système de RàPC.

Le but de ce projet est de formaliser le contenu des référentiels afin d'en permettre un accès intelligent pour les médecins. La diffusion de connaissances standardisées doit aboutir à une standardisation des pratiques en cancérologie et ainsi à une amélioration de la qualité des soins. Un problème cible décrit une patiente et sa maladie (âge, taille de tumeur, etc.), une solution Sol appartient à Solutions est une proposition de traitement.

Dans ce projet, la base de connaissances pour le traitement du cancer du sein est utilisé de façon directe pour la majorité des cas et doit être adaptée selon les principes du RàPC quand un problème de décision cible « non standard » est rencontré.

- **Le projet CHEF**

CHEF est un logiciel qui propose des recettes de cuisine en utilisant la technique du RàPC. L'utilisateur (cuisinier) indique à ce programme les ingrédients (aliments) dont il dispose et le programme CHEF cherche à élaborer une recette à partir de ces ingrédients

CHEF dispose d'une très grande base de "cas" qui sont en effet des recettes valides et tente de créer une nouvelle recette contenant les ingrédients disponibles. CHEF a la particularité de tenir compte des échecs et de les sauvegarder afin d'éviter de les reproduire. Lorsque les contraintes imposées par l'utilisateur sont trop fortes et qu'aucune solution n'a été trouvée, CHEF

donne une explication indiquant les raisons de l'échec.

- **Le projet AxIS**

Le projet AxIS est un projet pluridisciplinaire (Extraction de Connaissances à partir de données, Intelligence artificielle et Génie logiciel). Le but du projet est de développer des méthodes et des outils pour l'aide à la conception, l'évaluation et l'amélioration de systèmes d'information dirigées par les usages.

- **Le projet GO3**

« Nouvelles connaissances pour la sécurité » du PREDIT (Programme national de Recherche Et D'Innovation dans les Transports terrestres). Le projet a pour objectif de développer un système de raisonnement à partir de cas qui à partir d'un groupe d'accidents survenus dans un secteur géographique particulier, permettra d'y associer un profil de scénarios types et qui fournira un ensemble d'objectifs et de principes d'action à entreprendre pour aménager le secteur étudié.

- **Le projet Proteus**

Le projet Proteus est un projet Européen financé par le Ministère de l'économie et des finances sous la responsabilité de European Commission Initiative ITEA. Il a pour but de fournir une plate-forme et les concepts génériques pour la construction des systèmes de e-maintenance industrielle incluant les systèmes existants d'acquisition de données, de contrôle-commande, de gestion de la maintenance, d'aide au diagnostic, de gestion de la documentation...etc. Le but de ce projet est non seulement d'intégrer des outils existants, mais aussi de prévoir l'évolution de ceux-ci à travers d'introduire de nouveaux services.

Le site Web de ce projet est : <http://www.proteus-iteaproject.com>

- **Le projet Swale**

Le projet Swale cherche à fournir des explications à des situations anormales. Il apporte notamment des explications sur les causes de la mort des animaux ou des hommes. Le programme va par exemple comparer la mort inattendue d'un cheval de course très connu en pleine force de l'âge à la mort d'un cycliste due à une consommation excessive de produits dopants.

- **Le projet Persuader**

Persuader est un outil de gestion de conflits basé sur le raisonnement par cas. Il est fonctionné sur le principe de négociation/médiation. Il est capable de fournir des solutions documentées pour la résolution de problèmes de groupe. Persuader fait en sorte de pouvoir construire un règlement mutuellement convenu entre les différents acteurs de la dispute.

- **Le projet Resyn/RàPC**

Le système Resyn/RàPC [Lieber et al., 1996 ; Lieber 1997] est une application du RèPC pour la conception de plans de synthèse en chimie organique en interaction avec un chimiste. Le problème de la synthèse

organique est de concevoir un plan de synthèse (une séquence de réactions chimiques afin de construire une molécule appelée molécule cible, à partir de produits plus simples appelés précurseurs.

- **Le projet Déjà Vu**

Déjà Vu est un système de conception de logiciels de contrôle de véhicules robotisés industriels. Ces véhicules se déplacent dans une aciérie pour transporter des rouleaux et des bobines d'acier ainsi que leur chargement et déchargement.

- **Le projet PAD'IM**

Le projet Pad'IM (Poste d'Aide à la Décision IMMédiate) pour l'aide à la décision en supervision industrielle [Mille 1995b], [Mille 1995b], [Fuchs 1995].

9. Quelques rencontres sur le RàPC

Deux rencontres internationales s'organisent en alternance chaque année:

- **ECCBR (European Conference on Case-Based Reasoning)**

Cette conférence s'organise une fois tous les deux ans en Europe dans les années paires.

- **ICCBR (International Conference on Case Based Reasoning)**

Cette conférence internationale s'organise dans les années impaires. Cette année, c'est sa 7^{ième} version qui va être organisée à Belfast en Irlande du nord du 13 au 16 out 2007. La version précédente a déroulé à Chicago en Amérique en 2005.

Un site web est consacré à ces deux conférences (www.iccbr.org).

Cependant, un atelier RàPC s'organise chaque année en France depuis 1993. Cette année c'est sa 15^{ième} version qui se tiendra dans le cadre de la plate-forme AFIA 2007 qui aura lieu à Grenoble, du 2 au 6 juillet 2007. Le site web (<http://afia2007.imag.fr>). La version précédente (14^{ième}) c'est déroulée du 30 au 31 Mars 2006 à Besançon.

10. Conclusion

Nous avons présenté dans ce chapitre la technique du raisonnement à partir de cas qui représente le cœur de notre travail. Nous nous intéressons dans cette technique plus précisément à la phase de l'adaptation où nous allons utiliser une autre technique de l'intelligence artificielle, les réseaux de neurones, pour réaliser cette phase délicate.

Dans le chapitre suivant, nous présentons la technique des réseaux de neurones qui permet de réaliser des tâches de classification et c'est elle que nous allons l'utilisée par la suite dans notre contribution.

Chapitre III

Réseaux de Neurones

1. Introduction

L'idée des réseaux de neurones formels (ou réseaux neuromimétiques) vient de l'étude de cerveau humain. Il s'agit de simuler son fonctionnement sur des machines (ordinateurs, robots...etc). En réalité, ces réseaux se basent sur des modèles mathématiques et informatiques simplifiés appelés « neurones formels », qui permettent l'exécution de certaines fonctions du cerveau humain comme la mémorisation, l'apprentissage, le travail en parallèle, ...etc.

Cette technique de l'IA que nous présentons dans ce chapitre, nous allons l'utilisée par la suite dans notre travail, pour réaliser la phase qui nous intéresse dans les système de RàPC et qui est celle de l'adaptation des cas.

Ce chapitre sera organisé comme suite : tout d'abord, nous présentons un historique des recherches sur les réseaux de neurones. Ensuite, nous introduisons la relation entre le neurone biologique et le neurone formel (section 3). Dans la section 4, nous définirons qu'est ce qu'un réseau de neurones. Nous présentons les différentes architectures des réseaux de neurones dans la section 5. Après, nous décrivons une phase importante qui précède l'exploitation d'un réseau de neurones et qui est celle de l'apprentissage (section 6). Nous citons par la suite, les différentes étapes pour la mise en place d'un réseau de neurones (section 7). Ensuite, nous rappelons quelques célèbres réseaux de neurones (section 8). Dans les deux dernières sections (9 et 10), nous présentons respectivement quelques domaines d'application, et les avantages et les inconvénients de cette technique.

2. Historique

En 1943, Les neurologues Warren Sturgis McCulloch et Walter Pitts ont construit le premiers modèles du neurone artificiel, qui deviendra la base de toutes les autres recherches, à la suite de leur article fondateur : *What the frog's eye tells to the frog's brain*. Après (1949), le physiologiste américain Donald Hebb de l'université MC GILL de Montréal a élaboré la première règle d'apprentissage nommée règle de Hebb décrits dans son ouvrage *The Organization of Behaviour*. Cette règle permet de modifier la valeur des coefficients synaptiques en fonction de l'activité des unités qu'ils relient. Ensuite, Franck Rosenblatt en 1958 développa le premier réseau de neurones qui permet la bi-classification (le Perceptron). En 1969, Marvin Lee Minsky et Seymour Papert ont publié un ouvrage mettant en exergue quelques limitations théoriques du Perceptron, ce qui a provoqué un sommeil des travaux qui a duré environ quinze ans.

En 1982, le physicien John Joseph Hopfield, a remis à la mode le modèle connexionnisme en publiant un article introduisant un nouveau modèle de réseau de neurones : le modèle récurrent. Et en 1986, l'élaboration de la technique de la rétropropagation [Rumelhart et al., 1986] a lancé définitivement le monde des recherches, ce qui a provoqué donc la diffusion des réseaux de neurones dans différents domaines d'applications.

3. Du neurone biologique au neurone formel

Avant de décrire qu'est ce qu'un neurone formel (artificiel) et son principe de fonctionnement, nous décrivons d'abord le neurone biologique qui présente la base du neurone formel.

3.1. Neurone biologique

C'est l'organe élémentaire de base qui forme le système nerveux des êtres vivants, il a servi à la mise au point des premiers neurones formels (**Figure 3-1**).

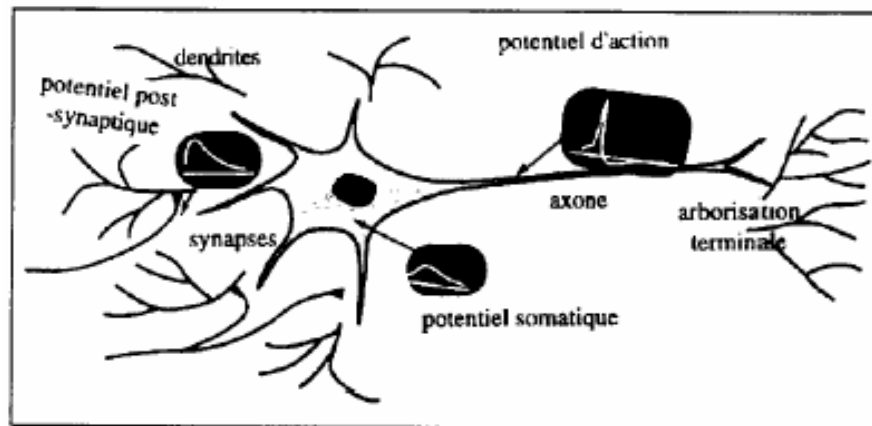


Figure 3- 1: Neurone biologique extrait de [Hérault et al., 1994]

3.1.1. Composants d'un neurone

Le neurone biologique est composé des éléments suivants :

- *Les dendrites* : se sont des extensions très ramifiées et très fine. Leur fonction est de recevoir des signaux (électriques) qui arrivent des autres neurones et de les transmettre au corps cellulaire du neurone;
- *Le corps cellulaire (noyaux)* : il présente le noyau du neurone. Sa tâche est de réaliser la sommation des signaux reçus en entrée et selon le résultat obtenu, il délivra un signal électrique en sortie ;
- *L'axone* : c'est un relais (conducteur de signaux) entre la sortie d'un neurone et l'entrée d'un autre.

Ainsi, le cerveau humain se constitue d'une multitude de neurones (plus de 100 milliards), reliés entre eux par l'intermédiaire d'axones et de dendrites. Ces derniers ne sont que des sortes de conducteurs d'électricité véhiculant des messages d'un neurone à l'autre. Les dendrites sont considérées comme des entrées alors que les axones jouent le rôle des sorties des neurones.

3.1.2. Fonctionnement

Le fonctionnement naturel d'un neurone tel qu'il est décrit par les biologistes, se consiste en général à recevoir et à émettre des signaux électriques, et cela comme suit :

Au niveau de chaque neurone, il y'a une intégration des signaux reçus (qui parviennent des autres neurones) au cours du temps. C'est en quelque sorte une sommation des signaux reçue qui se produit au niveau de chaque neurone. Quand cette somme dépasse un certain seuil, ce dernier (le neurone) émet à son tour un signal électrique aux neurones qui sont reliés à sa sortie (axone).

Ainsi les neurones communiquent en émettant des trains de potentiels rapides et très courts de l'ordre de quelques millisecondes.

3.1.3. Notion de synapse

Le neurone reçoit les signaux provenant d'autres neurones au niveau des dendrites qui sont le pôle récepteur de la cellule. Chaque branche de cette arborisation se termine par un point synaptique. La synapse est le lieu de connexion entre un axone et une dendrite (Figure 3-2).

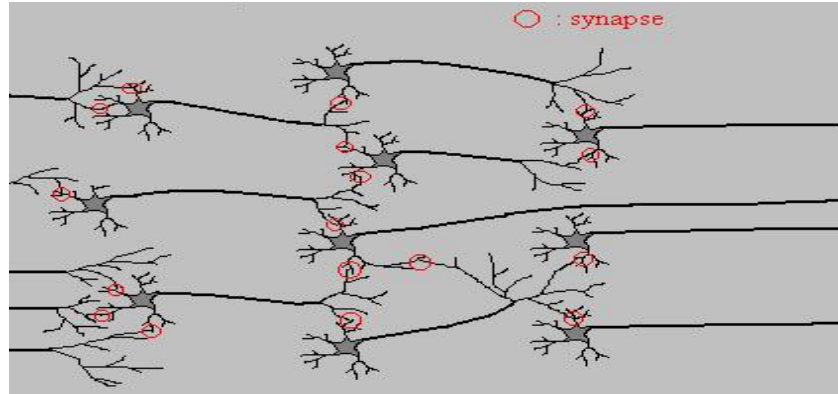


Figure 3- 2: Les synapses [Haykin 1994].

3.1.4. La transmission au niveau du synapse

Lorsque la modification du potentiel électrique atteint la terminaison de l'axone, les canaux des ions du calcium s'ouvrent. Après, les ions Ca^{++} entrent dans la synapse, et déclenchent des réactions en chaîne au niveau des vésicules synaptiques. Celles-ci se collent à la membrane cellulaire et libèrent des "neuromédiateurs". Ces derniers viennent ensuite s'incruster dans les récepteurs post-synaptiques de la dendrite de l'autre neurone. Cela provoque un changement de potentiel qui se propage le long de la dendrite vers le corps cellulaire (Figure 3-3). Suivant le type de la synapse, l'activité d'un neurone peut renforcer ou diminuer l'activité de ses voisins, on parle donc de synapse excitatrice ou inhibitrice [Milgram 1993].

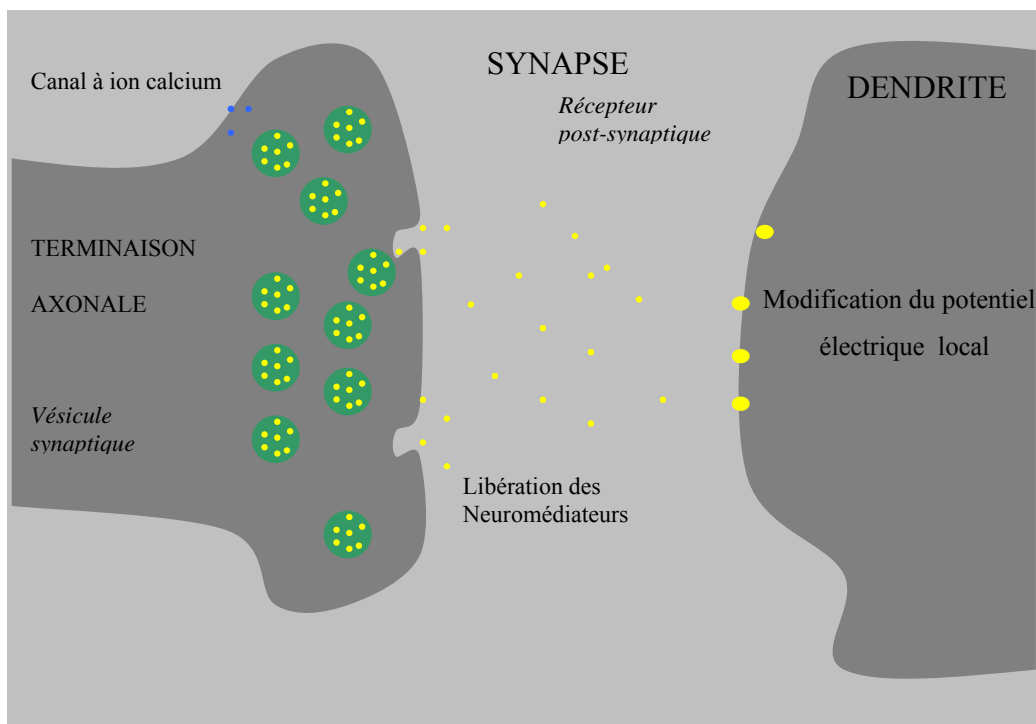


Figure 3- 3: Fonctionnement d'une synapse [Haykin 1994].

3.2. Neurone formel

3.2.1. Description

Le neurone formel est un modèle mathématique très simple dérivé d'une analyse de la réalité des neurones biologiques (Figure 3-4).

Pour le réaliser, l'intégration des signaux observés au niveau du neurone biologique est remplacée par une simple sommation des signaux arrivant au neurone. La sortie est obtenue par une simple comparaison de cette somme au seuil de déclenchement, par exemple elle peut être égale à 1 si la somme est supérieure au seuil et 0 si elle est inférieure.

Plus simple, il suffit de soustraire le seuil considéré à la somme des entrées, et de faire passer le résultat par la fonction de transfert. La sortie du neurone est le résultat obtenu après le transfert. Ainsi, on obtient une représentation des propriétés physique du neurone biologique.

Pour donner plus de puissance au modèle formel, on associe aux connexions synaptiques des poids qui sont en effet des valeurs réelles. Chaque entrée d'un neurone est alors multipliée par la valeur de la connexion synaptique associée.

En résumé, on peut dire que le neurone formel réalise tout simplement une somme pondérée de ces entrées par les poids des connexions, ajoute un seuil à cette somme et fait passer le résultat par une fonction de transfert pour obtenir sa sortie [Jodouin 1994].

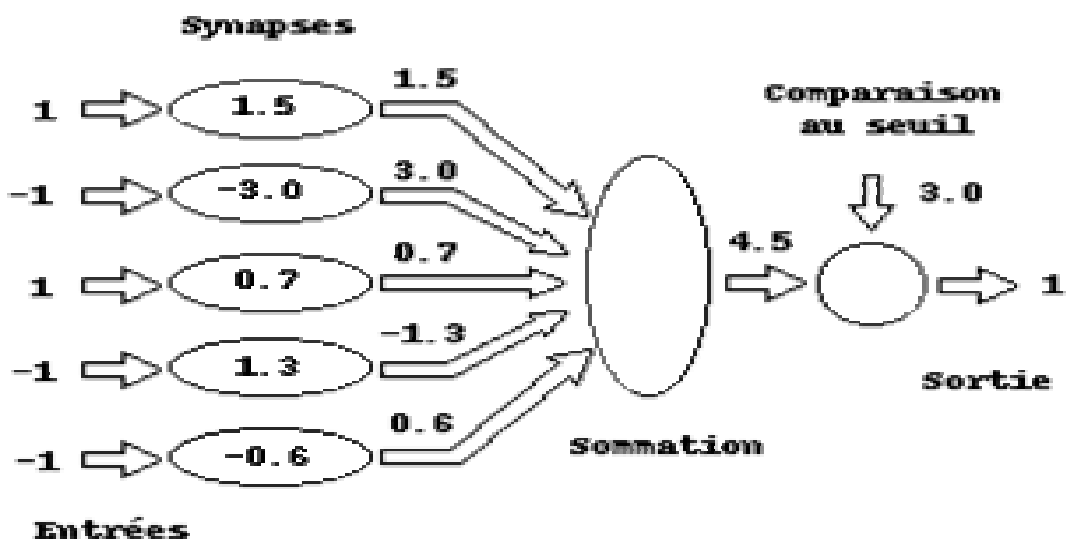


Figure 3- 4: Exemple du neurone formel.

3.2.2. Fonctionnement

Pour fonctionner, un neurone formel utilise des entrées qui sont des grandeurs réelles. Si on relie un neurone au monde extérieur par des capteurs, il peut réaliser une simple analyse de ce qu'il "perçoit". Si on représente les valeurs observées par le neurone sous forme d'un vecteur, le neurone réalise alors un découpage de son espace d'entrée (l'espace vectoriel auquel appartient le vecteur d'observation) en deux zones:

- **La zone d'activité:** dont les vecteurs donnent une sortie égale à 1 ;
- **La zone d'inactivité :** dont les vecteurs donnent une sortie égale à 0.

Comme le calcul effectué est en fait linéaire, la séparation l'est aussi. Les coefficients synaptiques et le seuil définissent l'équation d'un hyperplan qui est la frontière de la séparation entre les deux zones.

Mathématiquement, le neurone formel (Figure3-5) est considéré comme un automate ou une fonction non linéaire, bornée et dont la valeur dépend de certains paramètres appelés coefficients ou poids. Les entrées de cette fonction représentent les entrées du neurone, tandis que son résultat représente la sortie [Jodouin 1994].

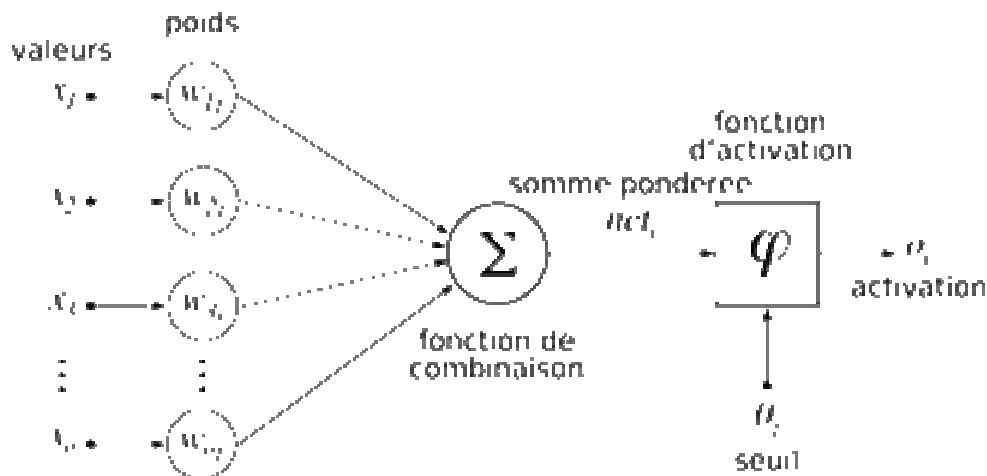


Figure 3- 5 : Neurone formel [Wikipédia].

Les X_i sont les n signaux d'entrée, et les W_{ij} les poids de lien reliant les neurones i et j .

L'activation pondérée « A » est donnée par :

$$A = \sum_{i=1}^n X_i W_{ij} = X_1 * W_{1j} + X_2 * W_{2j} + \dots + X_n * W_{nj}$$

La sortie finale est donnée par le seuil d'activation comme suit:

$$Y = F(A) = \begin{cases} 1 & \text{si } A \geq 0; \\ 0 & \text{si } A < 0; \end{cases}$$

3.2.3. Fonction d'activation

La fonction d'activation F détermine l'état interne du neurone en fonction de son entrée totale. Plusieurs modèles de cette fonction existent dans la littérature, on les décrit par la suite.

Fonction Heaviside définie par :

$$H(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases} \quad \text{(Figure 3-6)}$$

Fonction Signe définie par :

$$\text{Sign}(x) = \begin{cases} +1 & \text{si } x \geq 0 \\ -1 & \text{si } x \leq 0 \end{cases} \quad \text{(Figure 3-7)}$$

Fonction linéaire définie par :

$$F(x) = x \quad \text{(Figure 3-8)}$$

Fonction à saturation :

$$F(x) = \begin{cases} 1 & \text{si } x > k \\ -1 & \text{si } x < k \\ x/t & \text{sinon} \end{cases} \quad \text{(Figure 3-9)}$$

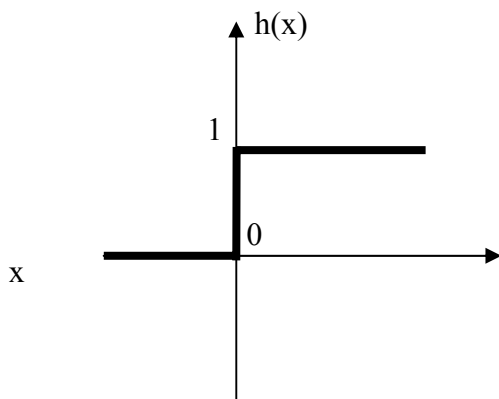


Figure 3- 6 : Fonctions Heaviside (à seuil)

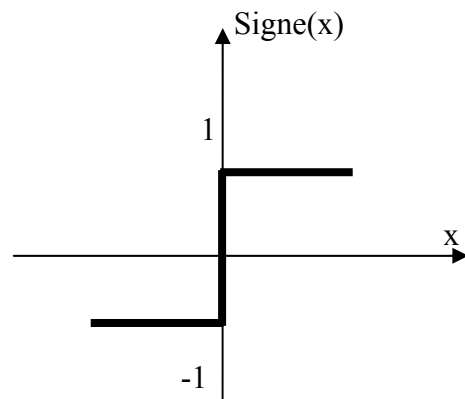


Figure 3- 7 : Fonctions signe

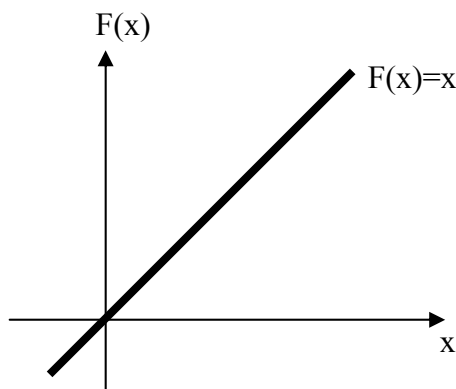


Figure 3- 8 : Fonction linéaire.

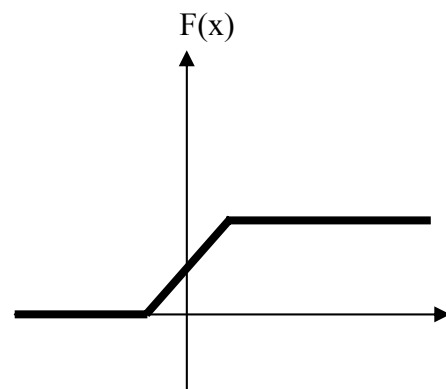


Figure 3- 9 : Fonction à saturation.

La fonction sigmoïde est définie par :

$$F(x) = \frac{1}{1 + e^{-x}}$$

La fonction gaussienne définie par :

$$g(x, w) = \exp\left(-\frac{\|x - w\|^2}{2\sigma^2}\right)$$

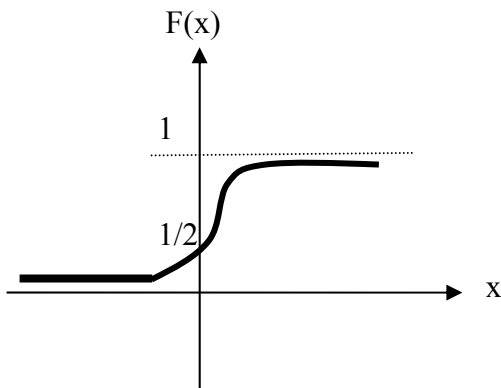


Figure 3- 10 : Fonction sigmoïde

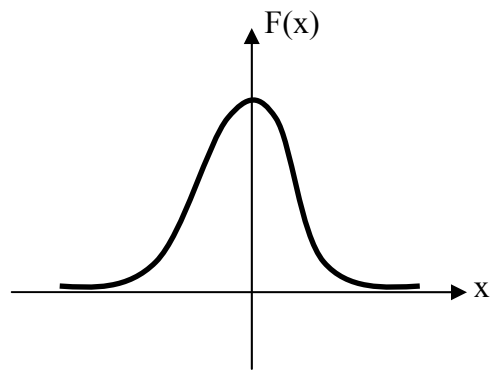


Figure 3- 11 : Fonction gaussienne

4. Réseaux de neurones

« Un réseau de neurones est un processus distribué de manière massivement parallèle, qui a une propension naturelle à mémoriser des connaissances de façon expérimentale et de les rendre disponibles pour utilisation. Il ressemble au cerveau en deux points:

1. la connaissance est acquise au travers d'un processus d'apprentissage;
2. les poids des connections entre les neurones sont utilisés pour mémoriser la connaissance. » [Haykin 1994]

Un réseau de neurones se compose de neurones qui sont interconnectés de façon à ce que la sortie d'un neurone puisse être l'entrée d'un ou plusieurs autres neurones. Ensuite il y a des entrées de l'extérieur et des sorties vers l'extérieur [Rumelhart et al., 1986].

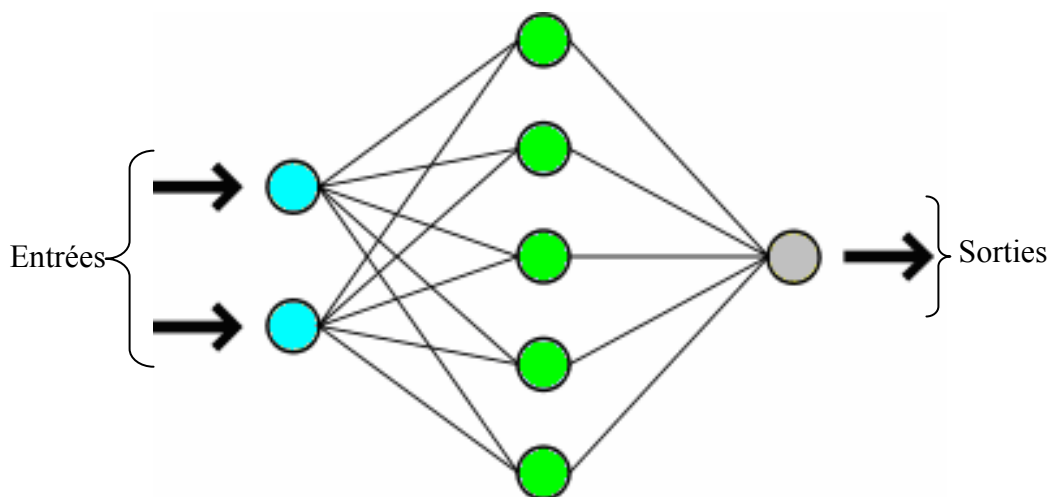


Figure 3- 11 : Exemple d'un réseau de neurones.

Formellement, un réseau neuromimétique est un graphe orienté et pondéré, dont les nœuds sont des unités de calcul que l'on appelle « neurones formels », et les arêtes se nomment « liens synaptiques » ou « connexions ». Ces derniers portent des connaissances de nature numérique [Jodouin 1994].

5. Architectures des réseaux de neurones

Les réseaux neuromimétiques peuvent se répartir en deux grandes catégories: la première est dite « réseaux à couches » (Architecture non- bouclée), et la seconde est dite « réseaux récurrents » (Architecture bouclée), qui est un peu plus complexe que la première.

La connectivité des réseaux peut être totale c'est à dire que tous les neurones sont connectés entre eux, ou bien par couches et dans ce cas là, les neurones d'une couche ne sont connectés qu'aux neurones de la couche suivante [Jodouin 1994].

5.1. Architecture en couches (feed-forward)

Les réseaux de neurones non bouclés sont des graphes dont les neurones sont connectés entre eux de telle sorte que l'information circule des entrées vers les sorties sans retour arrière, c'est à dire que graphiquement un réseau de cette architecture ne représente pas de circuits. À l'intérieur de cette classe on distingue les réseaux à deux couches et les multicouches. Ce genre de modèles est utilisé pour effectuer des tâches d'approximations, de modélisation, de classification...

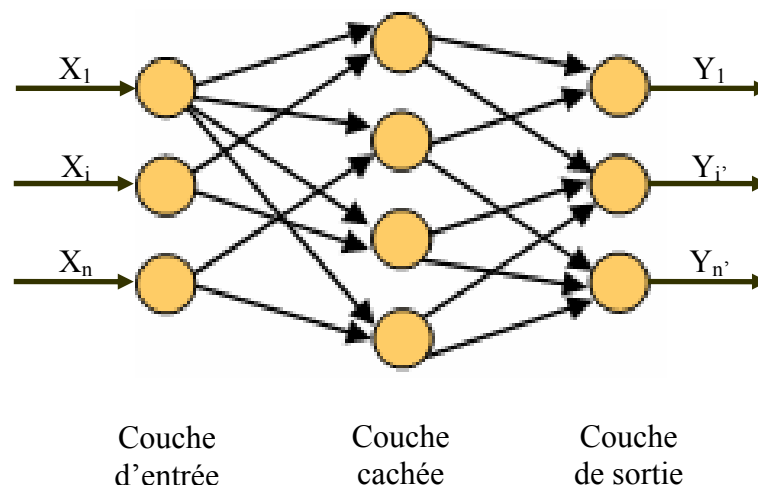


Figure 3- 12 : Réseaux feed-forward.

Parmi les réseaux de neurones qui sont basés sur ce type architecture, on trouve le Perceptron, le Perceptron multi-couches, le réseau RBF (Radial Basic Fonctions)

5.2. Architecture récurrente (feedback)

Dans ce type d'architecture, On trouve des boucles, soit au niveau des neurones, soit entre les couches. On distingue dans cette catégorie les réseaux à compétition et les réseaux à connexions symétriques, ils sont utilisés généralement dans la modélisation de systèmes dynamiques, la détection automatique des clusters [Jodouin 1994].

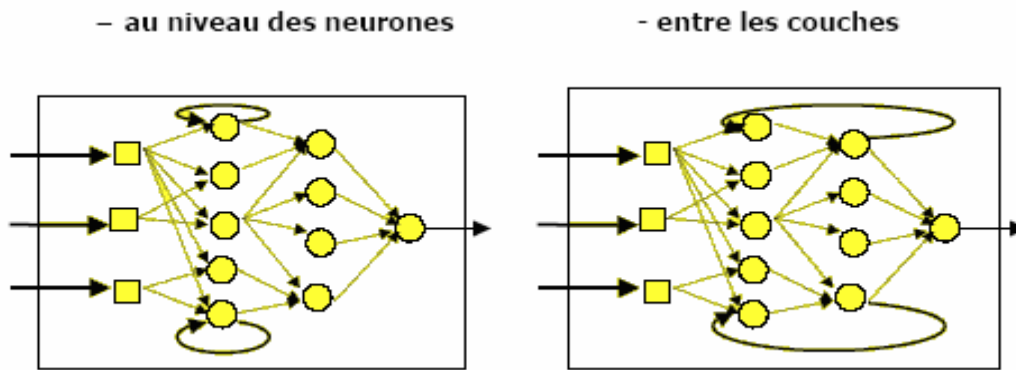


Figure 3- 13 : Réseaux feedback.

Parmi les réseaux de neurones qui sont basés sur ce type architecture, on trouve les cartes auto-organisatrices de Kohonen, les réseaux de Hopfield, les réseaux ART (Adaptative resonance theory)

6. Apprentissage des réseaux de neurones

6.1. Définition

Une fois qu'on a choisi l'architecture d'un réseau de neurones, et après qu'on initialise aléatoirement les poids des arcs du réseau, il est nécessaire d'effectuer un apprentissage pour déterminer les valeurs des poids permettant à la sortie du réseau de neurones d'être aussi proche que possible de l'objectif fixé.

On appelle « apprentissage » des réseaux de neurones, la procédure qui consiste à estimer les paramètres des neurones du réseau, afin que celui-ci remplisse au mieux la tâche qui lui est affectée.

Nous distinguons trois classes des techniques d'apprentissage, à savoir : apprentissages supervisés (supervised learning), apprentissages semi-supervisés (semi-supervised learning), et les apprentissages non-supervisés (unsupervised learning).

6.2. Apprentissage supervisé

Dans ce type d'apprentissage, on présente au réseau les entrées avec les sorties que l'on désire pour ces entrées (comportements de référence). Le réseau s'adapte par comparaison entre le résultat qu'il a calculé, en fonction des entrées fournies, et la réponse attendue en sortie [Luo et al., 1989]. Il va modifier ses poids jusqu'à ce que la sortie calculée soit très proche de la sortie réelle.

Ce type d'apprentissage est aussi appelé apprentissage par maître, car le réseau dans ce cas doit se reconfigurer, c'est à dire calculer ses poids afin que la sortie qu'il donne correspond bien à la sortie désirée. Parmi les réseaux qui utilisent ce type d'apprentissage on a : le Perceptron, l' Adaline, le PMC... [Jodouin 1994]

6.3. Apprentissage non supervisé (auto organisationnel)

Pour des réseaux à apprentissage non supervisé, on présente seulement des entrées au réseau (pas de sorties) et on le laisse évoluer jusqu'à ce qu'il s'auto-organise (càd se stabilise), en utilisant les lois régissant l'évolution des synapses. Ex : la carte de Kohonen, VQ... [Jodouin 1994].

L'apprentissage non supervisé est bien adapté à la modélisation des données complexes (images, sons...), généralement des données symboliques, où l'on possède des règles moins précises qui gouvernent le comportement de système à modéliser par les réseaux de neurones [Leroy 1996].

6.4. Apprentissage semi-supervisé

Le comportement de référence n'est pas disponible dans ce cas, mais il est possible d'obtenir des indications qualitatives (ex : correct/incorrect) sur les performances du réseau, citons par exemple le LVQ comme réseau utilisant ce type d'apprentissage [Jodouin 1994].

6.5. Apprentissage par renforcement

L'apprentissage par renforcement est un type d'apprentissage supervisé. Certains auteurs le considèrent comme un type d'apprentissage distinct, d'autres le considèrent comme un cas d'apprentissage semi-supervisé. Ce type d'apprentissage est présenté en détail dans le premier chapitre.

7. Mise en oeuvre d'un réseau de neurones

Pour concevoir un réseau de neurones, il faut suivre les étapes suivantes :

7.1. Préparation des données

La préparation des données consiste à déterminer les entrées et les sorties du réseau, constituer la base d'exemples, coder les entrées et enfin optimiser le jeu de données.

7.2. Détermination des paramètres

Plusieurs paramètres du réseau doivent être déterminés :

- *Architecture* : détermination du nombre de couches et du nombre de neurones de chaque couche (pour les réseaux feed-forward) ;
- *Fonction de sommation* : définition l'opérateur qui combine les poids des arcs W_i (somme, minimum, maximum, majorité, produit). La fonction la plus utilisée est la somme ;
- *Fonction de transfert* : détermination de la fonction à appliquer sur le résultat de la fonction de sommation (sigmoïde, linéaire, gaussienne). La fonction de transfert la plus utilisée est la sigmoïde.
- *Fonction de normalisation* : transformation de la sortie pour la rendre compréhensible par l'utilisateur.
- *Fonction de compétition* : détermination de la ou des sorties qui seront présentes à la couche suivante (la plus forte, les deux plus fortes, ...).
- *Calcul de l'erreur* : calcul de l'écart entre la sortie donnée et la sortie calculée (erreur moyenne, erreur absolue, erreur quadratique, sens de variation de l'erreur). La plus utilisée est l'erreur quadratique.

Il n'y a pas de règles précises pour déterminer les paramètres d'un réseau de neurones, tout dépend de la nature du problème à traiter et des informations disponibles. La majorité des paramètres sont déterminés d'une manière heuristique (par un ensemble de tests).

7.3. Phase d'apprentissage

L'apprentissage consiste à descendre le réseau de façon itérative en ajustant les poids à chaque passage selon le calcul d'erreur jusqu'à ce qu'il n'y ait plus d'amélioration. Pour cela, un algorithme de rétropropagation de l'erreur est mis en œuvre.

Le principe de l'algorithme de rétropropagation du gradient est le suivant :

- Initialiser la matrice des poids au hasard
- Choisir un exemple en entrée
- Propager le calcul de cette entrée à travers le réseau
- Calculer la sortie de cette entrée
- Mesurer l'erreur de prédiction par différence entre sortie réelle et sortie prévue
- Calculer la contribution d'un neurone à l'erreur à partir de la sortie
- Déterminer le signe de modification du poids
- Corriger les poids des neurones pour diminuer l'erreur

Le processus recommence ainsi, à partir du choix de l'exemple en entrée, jusqu'à ce qu'un taux d'erreur minimal soit atteint.

7.4. Phase de test

Au cours de cette phase, le réseau sera testé pour savoir s'il a maîtrisé et appris à donner des résultats acceptables. Si ce n'est pas le cas, il faut revoir les étapes précédentes (détermination des paramètres, apprentissage...).

7.5. Phase d'exploitation

A ce stade, le réseau de neurones sera prêt à être exploité et il doit être en mesure de fournir des sorties adéquates à des nouvelles entrées autres que celles d'apprentissage.

8. Quelques réseaux célèbres

Plusieurs réseaux de neurones sont utilisés surtout dans l'industrie. Nous présentons quelques uns :

8.1. Le Perceptron

Le Perceptron, conçu en 1962 par Rosenblatt [Figure 3-12], est historiquement le premier réseau de neurones artificiel. L'origine de son nom provient du fait qu'il étudiait le mécanisme de la perception visuelle.

Il possède deux couches dont la première couche (couche d'entrée) représente les exemples présentés au réseau et la couche finale les cellules de décision (les résultats du calcul).

Les poids des liaisons entre la couche d'entrée et la couche finale peuvent être modifiés, en utilisant la règle suivante :

$$w_{kj}(t+1) = w_{kj}(t) + \Delta w_{kj}(t)$$

$$\text{avec } \Delta w_{kj}(t) = \begin{cases} +h & \text{si } (d_k = +1) \text{ et } (S_k = -1) \text{ et } (X_j = 1) \\ -h & \text{si } (d_k = -1) \text{ et } (S_k = +1) \text{ et } (X_j = 1) \\ 0 & \text{sinon} \end{cases}$$

Avec h une constante positive, d_k la sortie désirée du neurone k , S_k sa sortie réelle, X_j la $j^{\text{ième}}$ entrée. Notons que ce type de réseau se contente seulement à résoudre les problèmes linéairement séparables, de plus il n'optimise pas les hyperplans séparateurs [Jodouin 1994].

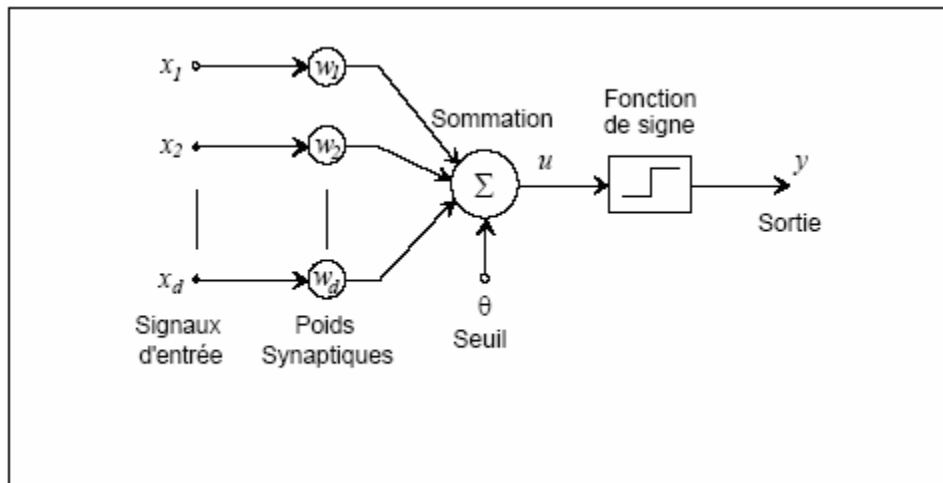


Figure 3- 14 : Le Perceptron de Rosenblatt [Gosselin 1996].

8.2. L'Adaline

L'ADAPtive LINear Element a la même architecture que le Perceptron, mais en revanche il possède des valeurs d'activations continues, une fonction d'activation linéaire et une règle d'apprentissage un peu développée que celle du Perceptron. Elle s'appelle la règle de Widrow & Hoff. En effet, cette loi minimise l'erreur quadratique en utilisant la méthode du gradient.

Contrairement à la règle du Perceptron, cette dernière opère sur les exemples bien classés et mal classés, en optimisant toujours les coefficients des hyperplans séparateurs.

Formellement on définit la dynamique ainsi que la règle d'apprentissage comme suit :

$$S_i = \sum W_{ij} \times X_j - \theta_i$$

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$$

$$\Delta W_{ij}(t) = h \times S_j \times (d_i - X_i)$$

Avec :

S_j : sortie réelle du neurone j ;

h : le pas d'apprentissage ;

X_i : la sortie réelle du neurone i ;

d_i : la sortie désirée du neurone i ;

θ_i : le seuil associé au neurone i ;
 t : numéro de l'itération [Jodouin 1994] .

8.3. Les Perceptrons multicouches (PMC ou MLP)

8.3.1. Présentation

Il représente une amélioration de l'adaline (**Figure3-15**), il s'agit d'un réseau comportant L couches, dont plusieurs couches intermédiaires sont dites cachées ; chaque neurone d'une couche étant totalement (ou partiellement) connecté aux neurones de la couche suivante.

On suppose que la rétine comporte un neurone particulier dont la sortie est constante (et qui est égale à 1), il est relié à l'entrée de tous les neurones des couches supérieures (les valeurs des connexions issues de ce neurone d'entrée particulier joueront le rôle des seuils pour les automates receveurs).

Chaque neurone k est un automate linéaire généralisé dont la fonction de transfert est supposée sigmoïde.

Pour modifier leurs poids les PMC utilisent, un algorithme de rétro propagation du gradient, où il est toujours question de minimiser l'erreur quadratique, en utilisant une fonction de transfert f dérivable (la sigmoïde).

La modification des poids est propagée de la couche de sortie jusqu'à la couche d'entrée.

Les PMC agissent comme un séparateur non linéaire et peuvent être utilisés pour la classification, la modélisation, le traitement de l'image ou l'aide à la décision [Jodouin 1994].

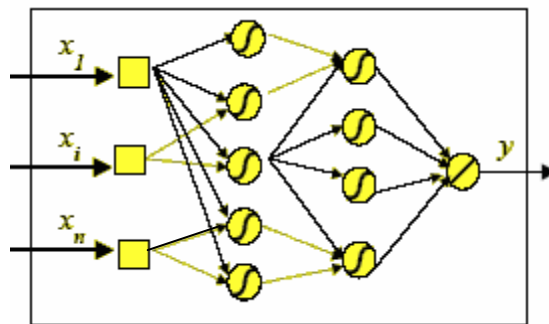


Figure 3- 15 : Perceptron multicouches

8.3.2. Algorithme de retro-propagation

En général les algorithmes d'optimisations utilisent la dérivée de la fonction considérée, donc son gradient. Lorsque les fonctions de transfert utilisées dans les neurones sont différentiables, l'erreur commise par un MLP sera aussi différentiable par rapport aux coefficients synaptiques du réseau, mais le problème qui se pose, c'est comment dériver cette erreur par rapport aux connexions qui ne sont pas directement reliées avec la couche de sortie.

Pour résoudre cet ennui on calcule les erreurs des neurones intermédiaires à partir des couches supérieures par une formule qui s'appelle rétro-propagation.

L'algorithme de rétro-propagation permet justement de calculer le gradient de cette

erreur de façon efficace : le nombre d'opérations (multiplication et addition) à faire est en effet proportionnel au nombre de connexions du réseau, comme dans le cas de la propagation des activations, ainsi cet algorithme rend possible l'apprentissage d'un PMC [Jodouin 1994].

8.3.3. Règle d'apprentissage

En considérant un réseau qui occupe m neurones en entrée et n neurones en sortie. Notons **net** la valeur d'entrée du neurone (la somme pondérée) et, **f** la fonction d'activation (la sigmoïde), la sortie d'un neurone a_i est donnée par :

$$a_i = f(\text{net}_i) = f\left[\sum_{j=1}^p w_{ij} * x_j\right]$$

Sachant que $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m)$ représente le patron d'entrée du réseau, et que $\mathbf{d} = (\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_n)$ indique les sorties désirées qui correspondent au vecteur des sorties effectives $\mathbf{s} = (\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n)$, on peut définir l'erreur quadratique qui lui est associée de la manière suivante :

$$E = \frac{1}{2} \sum_{j=1}^n (d_j - s_j)^2$$

Pour la minimiser on utilise la règle du Delta qui met à jour les poids :

$$w_{ij}(t+1) = w_{ij}(t) - h \cdot \text{err}_i \cdot a_j$$

Où err_i représente la responsabilité du neurone i dans l'erreur courante. Elle est calculée par rétro-propagation des erreurs de la couche, immédiatement supérieure à celle du neurone actuel.

Plus cette grandeur est petite, plus la valeur finale des poids est stable.

a_j : la sortie du neurone j , et h le pas d'apprentissage.

$\text{err}_i = -2 * (\mathbf{d}_i - \mathbf{s}_i) * f'(\text{li})$ pour la couche de sortie

$\text{err}_i = (\sum_k \text{err}_k * w_{ki}) * f'(\text{li})$ pour les couches cachés.

Où k parcourt tout les neurones vers lesquels le neurone i envoie une connexion.

li : est l'entrée pondérée (notée aussi **net_i**) du neurone i .

Note : la convergence du réseau dépend du pas choisie, il peut s'agir d'un petit pas (convergence lente), ou d'un pas élevé (risque d'oscillation) [Jodouin 1994].

8.3.4. Les applications du PMC

Nous avons choisi le PMC dans la reconnaissance des visages vu ses capacités dans le domaine de reconnaissance des formes, en effet plusieurs travaux utilisent ces outils dans des applications telles que :

- La reconnaissances des caractères (ou chiffres) manuscrits.
- La reconnaissance des empreintes digitales.
- la classification des pathologies cardiaques à partir des ECG.
- La reconnaissance de la parole.
- etc ...

8.4. Les réseaux RBF (Radial Basic Function)

Les réseaux **RBF** ont une architecture en couches, on trouve exactement : une couche d'entrée, une couche cachée, et une couche de sortie, ils ressemblent fortement aux PMC mais avec des différences importantes :

- Chaque neurone caché comporte une fonction noyau (gaussienne) au lieu de la sigmoïde.
- Pour un point donné de l'espace d'entrées, la sortie d'un neurone caché est égale à la hauteur de la gaussienne en ce point, elle est plus forte dans le centre du noyau.
- La sortie du réseau est une combinaison linéaire des sorties des neurones cachés.

L'utilisateur doit choisir le nombre de neurones cachés, l'algorithme d'apprentissage définit alors les paramètres des gaussiennes, et calcule les coefficients de la combinaison linéaire de sortie (c-à-d les poids entre la couche cachée et celle de sortie).

Un réseau RBF est plus facile à construire qu'un PMC, et apprend beaucoup plus vite, mais sur les problèmes difficiles, les réseaux RBF ne peuvent pas lutter avec les réseaux PMC, en terme de qualité des résultats [Jodouin 1994].

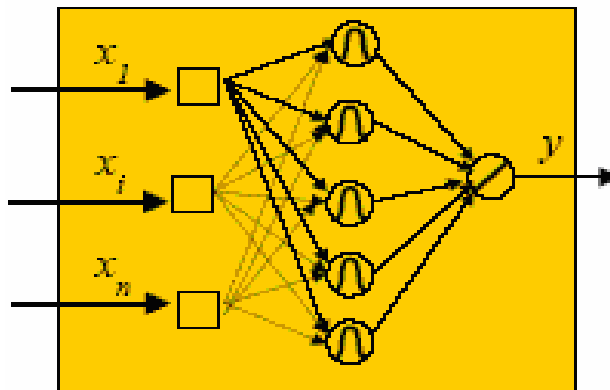


Figure 3- 16: Le réseau RBF.

8.5. Les réseaux de hopfield

8.5.1. Description

Développé en 1982, ce réseau est récurrent, et se base sur le principe des mémoires associatives, il est constitué d'une couche de neurones totalement connectées (symétriquement $W_{ij} = W_{ji}$).

Un réseau de Hopfield réalise une mémoire adressable par son contenu, il s'agit d'un réseau constitué de neurones de Mc Culloch et Pitts (à deux états, -1 et 1 ou 0 et 1), dont la loi d'apprentissage est la règle de Hebb (1949), qui s'inspire de la neurobiologie, son principe est qu'une liaison synaptique améliore son activité si et seulement si l'activité de ses deux neurones est corrélée, c'est à dire que le poids W_{ij} d'une connexion reliant un neurone i et un neurone j augmente quand les deux neurones sont activés en même temps, sinon le poids reste inchangé .

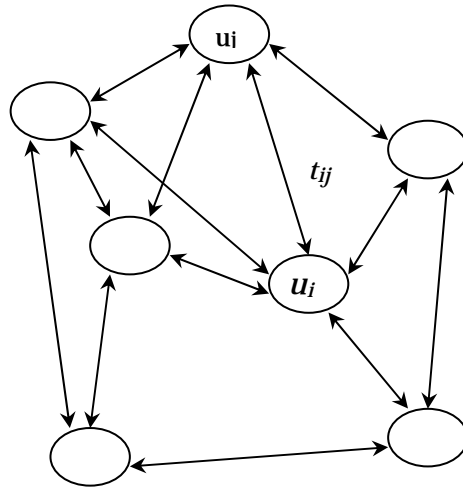


Figure 3- 17: Architecture du réseau de Hopfield.

8.5.2. Dynamique du réseau

Considérons un réseau contenant n neurones complètement connectés, à chaque neurone i on associe un état v_i (càd sa sortie) et on note t_{ik} la connexion entre deux neurones i et k , avec t la matrice des interconnexions (symétrique) et dont les éléments diagonaux t_{ii} sont nuls ; l'activation discrète d'un neurone i est donnée par :

$$v_i = h(u_i) = \{ 1 \text{ si } (u_i \geq 0), 0 \text{ sinon } \}$$

Avec h la fonction Heaviside définie précédemment.

$$u_i = \left(\sum_{k=1}^n t_{ik} * v_k \right) - \theta_i \quad \text{avec } k \neq i \text{ et } \theta_i \text{ le seuil associé au neurone } i$$

8.5.3. Fonction d'énergie

Soit, $\mathbf{I} = (v_1, v_2, v_3, \dots, v_n)$ l'état du réseau constitué par l'ensemble des états élémentaires de chaque neurones ; partant d'un état initial du réseau qui représente le problème, on veut atteindre un état stable qui correspond au résultat du calcul (solution du problème), tout en minimisant une fonction $E(\mathbf{I})$ appelée 'énergie totale

$$E(\mathbf{I}) = -1/2 \left(\sum_{i,j} w_{ij} * v_i * v_j \right);$$

Suivant la règle définie plus haut, l'évolution du réseau se fait à énergie décroissante, ainsi les états à énergie minimum sont des points fixes pour l'évolution du réseau, formellement la règle s'applique comme suit :

$$W_{ij}(t+1) = W_{ij}(t) + \mu * (v_i * v_j)$$

Avec μ le pas d'apprentissage [Jodouin 1994].

Remarque : Il existe une possibilité de calculer directement les valeurs des connexions à partir des exemples, sans utiliser l'algorithme itératif.

8.6. Réseaux de kohonen

Ces réseaux sont inspirés des observations biologiques du fonctionnement des systèmes nerveux de perception des mammifères.

La Figure 3-18 indique que le réseau est constitué de deux couches, la première contient les patrons d'entrées, la deuxième constitue les sorties associés, Chaque

neurone de cette dernière est connecté à tout les neurones d'entrées à travers n connexions qui portent les poids W_{ij} .

Ce modèle utilise un apprentissage compétitif, dont le principe est, qu'un seul neurone sera activé, après la propagation, ce dernier sera ensuite encouragé. (c'est-à-dire que le neurone gagnant tout seul, selon une certaine fonction, ainsi que, dans certains cas, un certain voisinage de ce nœud, verra son poids modifié).

On calcule le neurone gagnant en choisissant, celui dont le vecteur des poids est le plus proche du vecteur d'entrées selon une distance euclidienne. L'apprentissage consiste à rapprocher les poids des neurones gagnants vers les valeurs d'entrées. De cette façon, une application successive de plusieurs signaux d'entrées différents « organisera » les poids du réseau.

Les neurones d'une couche de Kohonen possèdent également des liaisons entre eux, dans le but de « compétition », en effet elles servent à définir le voisinage qui profitera de l'apprentissage. La taille et la manière de calculer le voisinage d'un neurone de la couche de Kohonen peuvent varier d'un réseau à l'autre. Ce voisinage peut aussi bien être nul, c'est-à-dire que seul le neurone activé est entraîné à chaque présentation d'un élément.

Les cartes de Kohonen se trouvent utiles dans le cas d'une classification automatique. En effet, on utilise les neurones de la couche de Kohonen comme des indicateurs d'appartenance à une classe donnée.

Pour un certain neurone, le vecteur \mathbf{W} de poids qui lui est associé indique le représentant de la classe que ce neurone a la tâche de symboliser. Ce neurone ne sera activé que si l'élément présenté au réseau est plus près du représentant de cette classe et que les représentants des autres classes sont plus ou moins loins. Bien sur, le problème de l'utilisation de réseaux de Kohonen pour effectuer ce type de classification automatique est dans le nombre de classes, et en même temps, dans le nombre de neurones de la couche de Kohonen. Notons que le réseau de Kohonen trouvera des classes même s'il n'en existe pas [Jodouin 1994].

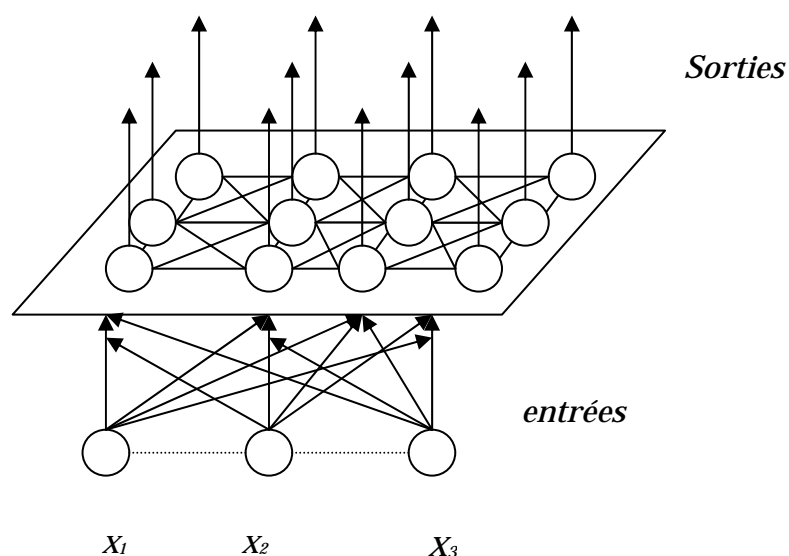


Figure 3- 18 : Architecture d'un réseau de kohonen.

8.6.1. Dynamique du réseau

L'activation du réseau de Kohonen par application du vecteur d'entrée X , revient à trouver le neurone gagnant j (recherche de la distance euclidienne minimale) :

$$j(p) = \text{MIN}_i \{ \| X - W_i(p) \| \}; \quad i=1,2,\dots, m.$$

Le i parcourt la couche de sortie et m représente le nombre de neurones de la couche de décision (sortie).

8.6.2. Apprentissage :

$$\Delta W_{ik}(p+1) = \begin{cases} \alpha \cdot [X_k - W_{ik}(p)], & i \in \Lambda_j(p) \\ 0, & i \notin \Lambda_j(p) \end{cases}$$

Où $\Lambda_j(p)$ est le voisinage du neurone gagnant j ;

P : le numéro de l'itération ;

K : parcourt tout les neurones d'entrée ;

W_i : Le vecteur des poids associé au neurone de sortie i ;

α : Le pas d'apprentissage [Jodouin 1994] ;

Il existe d'autres réseaux compétitifs tels que le VQ, LVQ, ART...

9. Domaines d'application

Les principales applications des réseaux de neurones sont l'aide à la décision, la planification, la compression de données, la prévision météorologique et la reconnaissance de formes. Les réseaux de neurones sont aussi utilisés par les cognitivistes et les neurobiologistes pour simuler le comportement humain.

L'étude des réseaux de neurones est une voie prometteuse de l'intelligence artificielle et dont les domaines d'applications sont nombreux :

- Industrie : contrôle qualité, diagnostic de panne, corrélations entre les données fournies par différents capteurs, analyse de la signature ou d'écriture manuscrite ;
- Finance : prévision et modélisation du marché, sélection d'investissements, attribution de crédits ;
- Télécommunications et informatique : analyse du signal, élimination du bruit, reconnaissance de formes (bruits, images, paroles), compression de données ;
- Environnement : évaluation des risques, analyse chimique, prévisions et modélisation météorologiques, gestion des ressources.

10. Avantages et inconvénients

L'avantage des réseaux de neurones est qu'ils acceptent des données incomplètes, incertaines ou bruitées. Ils présentent de plus une grande robustesse face aux défaillances techniques, et s'enrichissent de leurs expériences. Par contre, leur architecture parallèle nécessite l'emploi de processeurs spécialisés pour accélérer les

calculs. De plus, il est nécessaire de passer par la phase d'apprentissage avant d'utiliser le réseau. Enfin, il ne faut pas penser qu'un réseau de neurone est capable d'égaliser un cerveau humain, dont la structure est encore trop peu connue pour qu'on puisse le répliquer.

10.1. Autres avantages

- Apprentissage automatique des poids (Les règles d'apprentissage).
- Capacité de généralisation (La phase des testes)
- possibilité de faire le parallélisme (les éléments de chaque couche peuvent fonctionner en parallèle)
- Résistance aux pannes (si un neurone ne fonctionne plus, le réseau ne se perturbe pas) ;
- modélisation des fonctions inconnues
- bon approximateurs [Leroy 1996].

10.2. inconvénients

- Représentation complexe. (généralement l'architecture du réseau de neurones est complexe, et même si on la schématise, elle reste peu lisible)
- Paramètres difficiles à interpréter « physiquement » (boîte noire)
- inexistence des règles définissant l'architecture du réseau en fonction du problème à résoudre.
- les algorithmes d'apprentissage consomment un temps de calcul considérable [Leroy 1996].

11. Conclusion

L'intérêt porté aujourd'hui aux réseaux de neurones tient sa justification dans quelques propriétés fascinantes qu'ils possèdent (et qui sont cités dans les avantages). Ces outils ont permis aussi le dépassement des limites de l'informatique traditionnelle, en offrant des services variés tels que la classification, la modélisation, l'approximation...etc.

Dans le chapitre suivant, nous allons présenter un état de l'art des techniques d'adaptation et d'acquisition de connaissance d'adaptation avec une comparaison selon quelques caractéristiques.

Chapitre IV

Techniques d'adaptation (Etat de l'art)

1. Introduction

L'adaptation des cas dans les systèmes de RàPC nécessite l'utilisation d'une stratégie d'adaptation et l'acquisition d'un ensemble de connaissances (connaissances d'adaptation) qui vont être utilisées pour l'accomplissement de cette phase qui est la plus délicate dans le processus de RàPC.

Nous allons présenter dans la section suivante ce chapitre quelques techniques qui se sont intéressées à la phase de l'adaptation et à l'acquisition de connaissances associées (section 2) avec une comparaison selon quelques critères discriminants (section 3).

2. Quelques techniques d'adaptation en RàPC

Beaucoup de travaux se sont intéressés par la phase de l'adaptation et l'acquisition de connaissances associées. Nous allons présenter dans ce qui suit, quelques uns de ces travaux en citons le principe de sa technique d'adaptation, et son domaine d'application.

2.1. [Hammond 1989]

- *Principe* : (Hammond, 1989; Hammond, 1990) décrit une approche de planification à partir de cas. Trois types d'apprentissage sont décrits. Ils relèvent tous du learning by remembering, qui consiste à indexer en mémoire des expériences de résolution de problèmes de planification. Un de ces types « *the critic learning* » concerne l'ACA: il consiste en l'apprentissage automatique de moyens de réparer des échecs de plan.
- *Domaine d'application* : Cette technique est appliquée dans le système *Chef* pour la conception de recettes de cuisine. L'adaptation de *Chef* utilise un ensemble de règles d'adaptation. La base de cas de *Chef* contient des recettes déjà validées. Pour élaborer une nouvelle recette, *Chef* recherche dans sa base de cas, la recette ayant un maximum de caractéristiques en commun (les ingrédients, le mode de cuisine, les buts, les contraintes...etc.) avec la nouvelle situation (le problème cible). Une fois *Chef* a créé sa nouvelle recette, il va simuler son exécution avant de la fournir à l'utilisateur.

2.2. [Hastings et al., 1995]

- *Principe*: cette technique d'adaptation regroupe le RàPC et raisonnement à partir de modèles. L'adaptation utilise les influences des attributs de la partie problème sur les attributs de la partie solution. Si ces influences sont connues qualitativement grâce aux experts, leurs valeurs numériques sont estimées par minimisation de l'erreur quadratique moyenne sur un échantillon de cas entre la prédiction effectuée par le système paramétré par ces valeurs et la prédiction d'un expert. Cette ACA consiste donc à raffiner le processus d'adaptation en apprenant ses paramètres.
- *Domaine d'application*: le système *CARMA* pour la prédiction des dégâts causés par les invasions de sauterelles. *CARMA* utilise la méthode de K-plus proche voisins et deux autres approches inductives (les arbres de décision et l'approximation linéaire) pour la recherche de cas similaires.

2.3. [Hanney et al., 1996]

- *Principe*: Cette technique permet une acquisition automatique de connaissances d'adaptation. Elle utilise la base de cas pour générer des règles d'adaptation à partir de couples de cas similaires. Ces règles ont comme prémisse la variation entre les problèmes comparés et comme conclusion la variation entre les solutions comparées. Le principe est d'interpréter la différence entre deux cas similaires de la base sous forme de règles d'adaptation, qui seront généralisées par la suite. Une mesure de confiance est associée à chaque règle en fonction de son degré de généralisation.
- *Domaine d'application*: cette technique est testée sur deux domaines, à savoir : (1) l'estimation du prix d'un logement, (2) la prédiction du temps de montée d'un servomécanisme.

2.4. [Smyth 95; Smyth et al., 1998]

- *Principe*: les auteurs présentent une technique de remémoration guidée par l'adaptation « *adaptation-guided retrieval* ». Le cas retenu dans la phase de remémoration est celui qui nécessite le moindre effort pour être adapté. L'argument est que ce ne sont pas toujours les cas sources les plus similaires au cas cible qui sont les plus faciles à adapter, en particulier lorsque la mesure de similarité s'appuie sur des caractéristiques de surface. Cette technique suggère de préparer et penser à l'adaptation dès la phase de remémoration pour quelle soit simplifiée.
- *Domaine d'application*: elle est appliquée au système nommé « *Déjà Vu* » pour la conception de logiciels de pilotage de véhicules robotisés industriels. Ces logiciels de control permettent de régulariser les actions des véhicules autonomes dans un environnement industriel réel (une aciérie). Ces véhicules chargent et déchargent des bobines d'acier.

2.5. [Leake et al., 1997]

- *Principe*: cette technique est semblable à celle de Smyth, elle suppose qu'une bonne remémoration d'un cas permet de réduire énormément l'effort d'adaptation. Pour cela, elle ajoute à la mesure de similarité la notion de "coût d'adaptation". Dans la phase de remémoration, elle utilise tout d'abord une mesure de similarité classique, ensuite, les cas sélectionnés seront favorisés selon leur coût d'adaptation. Cette technique d'adaptation s'appuie sur des règles et des cas d'adaptation. Un cas d'adaptation est constitué du triplet (Srce, Sol(Srce), Cible) et des connaissances mises en œuvre pour le résoudre.
- *Domaine d'application*: elle est appliquée dans le système DIAL pour la proposition de plans de réaction à des désastres naturelles et causés par de humains (séismes, crues, ...etc.) et d'évacuation des victimes. Une représentation conceptuelle du nouveau problème (le nouveau désastre), qui décrit les événements initiaux de ce désastre, est donnée par le système DIAL. Selon cette représentation, DIAL recherche dans sa base, une situation similaire à la situation courante et adapte son plan d'évacuation.

2.6. [Wilke 1997]

- *Principe* : l'auteur propose un cadre général pour l'ACA automatique s'appuyant sur l'approche « Knowledge light » qui est basée sur l'utilisation de connaissances déjà présentes dans le système RàPC. La notion de réservoirs de connaissances « Knowledge containers » est introduite. Quatre réservoirs de connaissances sont distingués : le vocabulaire (connaissances du domaine), les cas, les connaissances de remémoration (similarité) et les connaissances d'adaptation. Ces quatre réservoirs sont exploités par un algorithme d'apprentissage qui va les transmettre en connaissance d'adaptation.
- *Domaine d'application* : cette technique est appliquée sur une étude de cas d'une approche « Knowledge light » pour ACA.

2.7. [Purvis et al., 1998]

- *Principe* : Cette technique permet une combinaison de cas pour étendre son domaine d'application sur une large gamme de problèmes. Elle utilise *le problème de satisfaction de contraintes* pour effectuer cette combinaison qui se fait pendant la phase de l'adaptation. La solution d'un nouveau problème se construit en satisfaisant les nouvelles contraintes et en transformant des solutions mémorisées. L'algorithme appelé algorithmes de conflits minimal est appliqué pour aboutir à une représentation de cas commune.
- *Domaine d'application* : le système « COMPOSER » qui comporte deux domaines d'ingénierie de conception : (1) conception de séquences assemblées, (2) conception de configuration. COMPOSER fait la correspondance entre l'ancien et le nouveau problème en utilisant une structure de correspondance et la technique de plus proches voisins.

2.8. [Bergmann et al., 1995 ; Bergmann et al., 1996]

- *Principe* : les auteurs présentent une technique d'adaptation hiérarchique. Pour cela, des cas abstraits (des cas qui permettent de voir le problème à différents niveaux d'abstraction) sont utilisés au lieu des cas concrets et un raffinement des solutions aux différents niveaux d'abstraction permet de construire une solution au nouveau problème.
- *Domaine d'application*: cette technique est testée dans le domaine de planification de processus pour l'ingénierie mécanique : le système PARIS (Plan Abstraction and Refinement in an Integrated System). Ce système inclut une approche basée sur l'explication pour généraliser les cas durant leur apprentissages et de les spécialiser pendant leur résolution.

2.9. [Anand et al., 1998]

- *Principe*: cette technique permet une construction automatique de la base de cas et des règles d'adaptation à partir d'une base de données en utilisant les techniques du Data Mining. Des regroupements dans cette base permettent de construire des concepts dans lesquels sont choisis des cas représentatifs, qui forment la base de cas. Les connaissances d'adaptation sont construites à partir de cette même base de données grâce à des techniques de génération de règles, en particulier un algorithme de découverte de règles d'association.

- *Domaine d'application*: cette technique est appliquée dans le domaine de l'estimation des prix des logements selon leurs caractéristiques (endroit, surface, date de construction, nombres de pièces,...etc.).

2.10. [de Silva Garza et al., 1999]

- *Principe* : cette technique utilise les algorithmes génétiques pour effectuer l'adaptation. Les cas sélectionnés lors de la phase de remémoration forment la population initiale pour les *algorithmes* génétiques. Deux types d'adaptation sont utilisés pour faire évoluer le contenu des cas remémorés jusqu'à obtenir une solution satisfaisant les contraintes du domaine d'application. : (1) structurelle : par combinaison ou croisement, (2) paramétrique : par mutation. Durant la phase de remémoration, GENCAD retient 12 cas pour former la population initiale, et son cycle de l'AG est limité à 500 pour l'obtention de solution satisfaisante. Si le système atteint se nombre d'itération sans trouver une solution satisfaisante, il sera stoppé.
- *Domaine d'application*: cette technique est testée dans le système GENCAD pour la conception de tours et la disposition des résidences selon l'art de conception chinois « fong shui ». GENCAD est développé avec le langage Common LISP

2.11. [Corchado et al., 2001]

- *Principe*: Cette technique d'adaptation utilise un réseau de neurones artificiels de type RBF (*radial basis function*) pour réaliser la tâche d'adaptation. L'apprentissage du réseau est réalisé par un ensemble de cas sources similaires au nouveau problème, et qui sont sélectionnés lors de la phase de remémoration (par la méthode des k plus proches voisins). Le problème cible sera représenté à l'entrée du réseau déjà entraîné, et sa solution est récupérée à sa sortie. Cette technique effectue bien l'acquisition de connaissances d'adaptation. La fonction d'adaptation est bien maîtrisée par le réseau de neurones grâce aux cas sources sélectionnés dans la phase de remémoration, utilisés dans l'apprentissage du réseau. Le problème majeur de cette technique est qu'après chaque adaptation, aucune connaissance d'adaptation ne sera retenue par le système. Autrement dit, un réseau de neurones ne sert qu'à résoudre un problème donné et il sera rejeté.
- *Domaine d'application*: cette technique est testée dans le domaine de la prévision de la température de l'océan à une profondeur fixe, pour les prochains kilomètres parcourus par un navire. Les mesures de température sont réalisées dans l'océan atlantique entre la grande Bretagne et l'Islande.

2.12. [Jarmulak et al., 2001]

- *Principe* : (Jarmulak et al., 2001) décrit une approche d'ACA automatique à partir de la base de cas. Plusieurs méthodes de "knowledge-light" qui sont basées sur le principe d'adaptation par substitution sont développées pour permettre une ACA à partir de la base de cas. Comme [Leake et al., 1997], cette technique permet d'apprendre des cas d'adaptation, plutôt que des règles d'adaptation. Un cas d'adaptation décrit à partir de deux cas sources contient une représentation des dissimilarités entre leurs problèmes et leurs solutions, ainsi que le problème et la solution d'un des deux, considéré

comme le cas de référence.

- *Domaine d'application* : Elle est *appliquée* et testée sur un problème de conception de comprimés médicaux (quelle proportion des ingrédients, produit actif, excipient, etc. choisir pour un nouveau médicament?)

2.13. [Lieber et al., 2001; Lieber et al., 2003]

- *Principe*: cette technique permet l'ACA auprès d'experts, elle s'appuie sur des *comptes-rendus* informels d'adaptations effectuées par des experts et permet de mettre en évidence des schémas d'adaptation. La notion de "chemins de similarité" est introduite dans cette contribution comme un outil d'aide à l'acquisition et à la modélisation des connaissances d'adaptation. Un apport de ce travail est la décomposition d'une adaptation complexe en adaptations simples et généralisables.
- *Domaine d'application*: le projet KASIMIR d'aide à la décision en cancérologie du sein où un problème décrit les caractéristiques d'une patiente (age, taille de tumeur, ...etc.) et la solution donnée par le système KASIMIR est une proposition du traitement.

2.14. [d'Aquin et al., 2004]

- *Principe*: les auteurs présentent une technique d'apprentissage automatique des connaissances d'adaptation pour un système de RàPC. Cet apprentissage se fait à partir de la base de cas. L'idée est d'interpréter les différences entre les cas de la base, comme le résultat d'un processus d'adaptation. La technique d'apprentissage utilisée est l'extraction de motifs fréquents, technique de fouille de données qui suppose un formatage sous la forme d'un ensemble de propriétés booléennes.
- *Domaine d'application*: Elle est testée dans le projet KASIMIR utilisé par [Lieber et al., 2001] qui utilise un formalisme proche des logiques de descriptions.

2.15. Autres travaux

Nous ajoutons d'autres techniques (à titre de références) que nous manquons de documentation sur eux :

- [Koehler, 1996] utilise la logique de description pour l'adaptation des cas.
- [Cho et al., 1999] : cette technique est testée dans le domaine de l'estimation du prix d'un voyage.
- [Lee 2003] : cette technique est appliqué dans le domaine de l'estimation du prix d'une automobile occasion selon quelques caractéristiques (marque, puissance, année, kilométrage...etc.).

3. Comparaison entre ces travaux

Nous allons comparer les techniques d'adaptation et d'acquisition de connaissances d'adaptation présentées ci-dessus selon les critères suivants, à savoir :

- La nature de la technique (automatique ou supervisée) et la source de connaissances utilisée par le processus d'ACA;

- La manière de représentation des connaissances d’adaptation acquis durant le processus d’ACA.

3.1. Nature de la technique

La plupart des techniques présentées ci-dessus sont automatique « Knowledge light », c'est-à-dire qu’elles utilisent les connaissances déjà présentes dans le système de RàPC pour générer des connaissances d’adaptation. Ces connaissances sont utilisées soit pour:

- Fournir des couples de cas sources similaires qui vont être utilisés pour générer des règles d’adaptation, des cas d’adaptation, des dépendances des attributs de la partie problème sur ceux de la partie solution...etc.
- Fournir des cas sources qui vont être comparés à un problème cible un à un pour générer des connaissances d’adaptation.
- Fournir des connaissances d’adaptation déjà acquise, pour extraire à partir d’eux d’autres connaissances d’adaptation et de les généralisées.

D’autres sources de connaissances sont distinguées, telles que l’expert du domaine, les connaissances de remémoration... etc.

Le tableau suivant montre la nature et la source de connaissances utilisée pour chacune des techniques présentées dans la section 2 :

	Automatique	Supervisée	Source de connaissances
[Hammond 1989]	*		Les cas
[Hastings et al., 1995]		*	L’expert
[Hanney et al., 1996]	*		Les couples de cas
[Smyth 95; Smyth et al., 1998]	*		Les cas
[Leake et al., 1997]	*		Les cas d’adaptation
[Wilke 1997]	*		Les cas + CA ² + CR ³ + connaissances du domaine
[Purvis et al., 1998]	*		Les cas
[Bergmann et al., 1995 ; Bergmann et al., 1998]	*		Les cas
[Anand et al., 1998]	*		Une base de données
[de Silva Garza et al., 1999]	*		Les cas
[Corchado et al. 2001]	*		Les cas

² CA : Connaissances d’adaptation

³ CR : Connaissances de remémoration

[Jarmulak et al., 2001]	*		Les cas + CA + CR + connaissances du domaine
[Lieber et al. 2001; Lieber et al. 2003]		*	L'expert
[d'Aquin et al., 2004]	*		Les couples de cas

Tableau 4- 1 : Comparaison selon la nature et la source de connaissances

3.2. Type de connaissances acquises

Les connaissances d'adaptation peuvent prendre plusieurs formes telles que : des cas d'adaptation, des règles d'adaptation, des poids du réseau de neurones, le nombre de cas remémorés utilisés pendant l'adaptation,...etc.

Chacune des techniques présentées dans la section 2, représente les connaissances d'adaptation en utilisant une ou plusieurs de ces formes citées ci-dessus, mais la plus part d'elles représentent ces connaissances sous forme de règle d'adaptation. Le tableau suivant présente une comparaison de ces techniques selon la manière de représentation des connaissances d'adaptation acquises au cours du processus d'acquisition.

	Cas d'adaptation	Règles d'adaptation	Autres
[Hammond 1989]		*	
[Hastings et al., 1995]		*	
[Hanney et al., 1996]		*	
[Smyth 95; Smyth et al., 1998]			
[Leake et al., 1997]			*
[Wilke 1997]			
[Purvis et al., 1998]		*	
[Bergmann et al., 1995 ; Bergmann et al., 1998]			
[Anand et al., 1998]		*	
[de Silva Garza et al., 1999]			*
[Corchado et al. 2001]			*
[Jarmulak et al., 2001]	*		
[Lieber et al. 2001; Lieber et al. 2003]			*
[d'Aquin et al., 2004]		*	

Tableau 4- 2 : Comparaison selon le type de connaissances acquises

4. Conclusion

Nous avons présenté, dans ce chapitre, plusieurs travaux qui se sont intéressés aux problèmes que pose la phase de l'adaptation et d'acquisition de connaissances associées. Une comparaison entre les techniques présentées dans ces travaux est aussi présentée.

Dans le chapitre suivant, nous allons présenter notre contribution qui consiste à proposer une nouvelle technique qui permet l'adaptation des cas et, en même temps, l'acquisition de connaissances nécessaire pour l'accomplissement de cette phase délicate. Un cas d'expérimentation de cette nouvelle technique sera aussi présenté en vue de la tester sur un cas réel d'application.

Chapitre V

Notre contribution

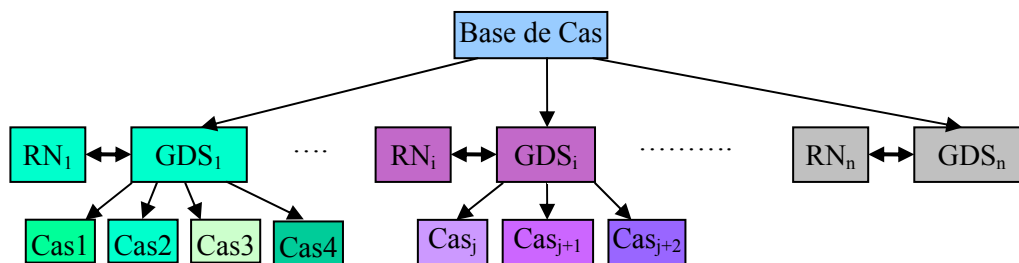
1. Introduction

Dans ce chapitre, nous allons présenter notre contribution qui consiste à proposer une nouvelle technique d'adaptation des cas dans les systèmes RàPC. La section suivante sera consacrée pour la description de cette technique (section 2). Dans la section 3, un domaine d'application sera tout d'abord présenté pour tester, par la suite cette technique sur ce domaine.

2. Technique proposée

Nous proposons une technique d'adaptation des cas qui est basée sur l'utilisation des réseaux de neurones. Pour cela, nous décomposons d'abord la base de cas en Groupes De Similarité (GDS). Nous associons ensuite à chaque GDS un réseau de neurone qui va être entraîné dans la phase d'apprentissage en utilisant les cas du groupe associé. Nous aurons donc autant de réseaux de neurones qu'il y'a de GDS.

La Figure 5-1 ci-dessous illustre l'organisation arborescente de notre base de cas.



GDS: Groupe De Similarité
RN : Réseau de Neurones

Figure 5-1 : Décomposition de la base de cas

2.1. Création des GDS

Pour la création des GDS, nous proposons d'utiliser un sous groupe d'attributs de la partie problème, jugés les plus discriminants, pour augmenter la ressemblance entre les cas d'un même GDS et aboutir à des cas bien adaptés dans la phase d'exploitation des réseaux de neurones. Le choix de ces attributs va influencer sur la qualité du système, d'où la nécessité de les sélectionner soigneusement pour obtenir des résultats pertinents.

2.2. Sélection des cas pivots

Après la création des GDS, nous suggérons de sélectionner un cas pivot pour chaque GDS pour réduire l'effort de la remémoration. Ainsi, au lieu de mesurer la similarité du nouveau problème avec tous les cas de la base, nous allons le comparer seulement avec les cas pivots. Nous proposons deux façons pour la sélection des cas pivots:

- Soit d'une manière automatique par le système RàPC en calculant la moyenne de chaque attribut des cas d'un même GDS, et le cas qui coïncide ou le plus proche du vecteur formé des moyennes calculées sera choisi comme cas pivot ;
- Soit manuellement en laissant le choix à l'utilisateur.

- Soit d'une manière semi-automatique en laissant le système sélectionner automatiquement un sous ensemble de cas, et l'utilisateur va faire son choix parmi ces cas.

L'idée d'utiliser des cas pivots vient du constat que dans notre cas, si un problème est similaire à deux cas qui appartiennent au même GDS, le résultat de l'adaptation sera le même puisque les deux cas ont un même réseau de neurone qui va faire l'adaptation. Cependant, après l'ajout de nouveaux cas à un GDS, nous proposons de recalculer son cas pivot.

2.3. Apprentissage des réseaux de neurones

Après avoir partagé la base de cas en groupes de similarité, nous allons créer un réseau de neurone de type PMC (Perceptron Multi-couches) pour chacun de ces groupes. Ensuite, un apprentissage supervisé de chacun de ces réseaux, par les cas que contient le groupe associé, est réalisé. Le résultat obtenu par le réseau va être comparé avec la partie solution de ce même problème. En calculant l'écart entre eux, ce dernier permettra de modifier les poids des arcs du réseau. Ce processus est répété jusqu'à ce que l'écart soit inférieur à un seuil fixé par l'utilisateur ou bien le nombre d'itérations atteint une certaine limite fixée par l'utilisateur.

La **Figure 5-2** ci-dessous illustre le déroulement du processus décrit précédemment sur un exemple.

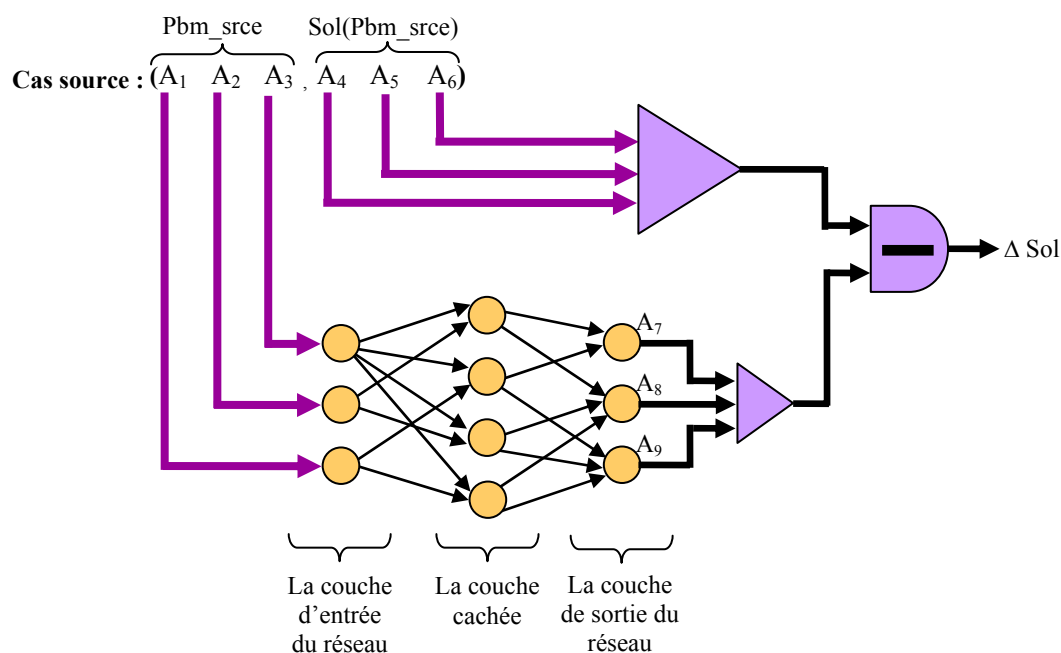


Figure 5- 2 : Fonction d'adaptation

Etant donné un cas source: ($pbm_srce, Sol(pbm_srce)$), et que pbm_srce contient les attributs A_1, A_2 et A_3 et la $Sol(pbm_srce)$ les attributs A_4, A_5 et A_6 . Les attributs de la partie pbm_srce (A_1, A_2, A_3) seront présentés à l'entrée du réseau de neurones associé au GDS affecté au problème en cours. Après calcul (propagation), le réseau de neurones va fournir une solution (A_7, A_8, A_9). En comparant cette dernière avec la solution réelle [$Sol(pbm_srce)=(A_4, A_5, A_6)$], on obtient une différence $\Delta Sol=(A_4-A_7, A_5-A_8, A_6-A_9)$. Les poids des arcs du réseau de neurones seront raffinés à partir de ΔSol et ajustés par la technique de rétro-propagation du gradient en vue d'optimiser

l'adaptation.

Après cette étape, les réseaux seront stockés dans une base de données et seront prêts à être utilisés par le système RÀPC.

2.4. Processus du raisonnement

Nous décrirons ci-dessous les trois principales phases qui se trouvent au coeur du processus RÀPC:

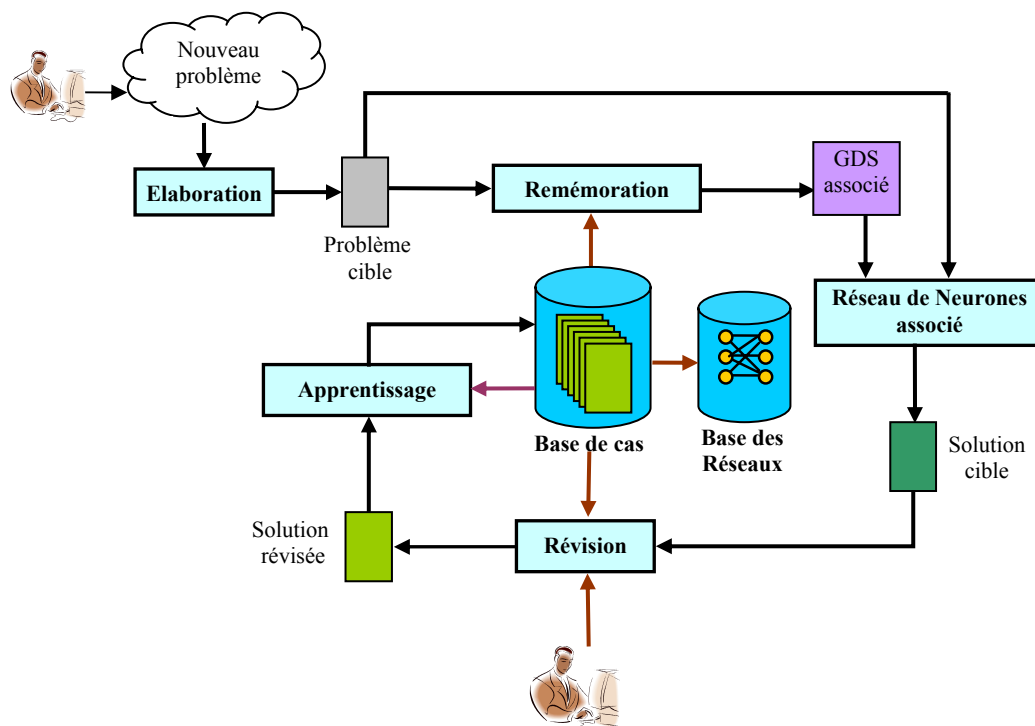


Figure 5- 3 : Notre processus RàPC

- La phase de remémoration, qui vient après la phase d'élaboration, permet d'affecter le nouveau problème au GDS du cas pivot le plus proche. Si la distance entre le nouveau problème et le cas pivot le plus proche est supérieure à un seuil donné, le nouveau problème sera rangé, s'il est jugé pertinent, dans un groupe spécifique (dont les cas qui le composent ne sont pas forcément similaires). Ce groupe spécifique sera utilisé ultérieurement comme nouvelle base qui sera décomposée à son tour en GDS.
- Pendant la phase d'adaptation, le nouveau problème sera présenté alors à l'entrée du réseau de neurones associé au GDS du cas pivot le plus proche. Après calcul, le réseau fournira en sortie la solution adaptée (Figure 5-3). Dans notre cas, la fonction d'adaptation qui se résume au réseau de neurones, n'utilise comme entrée que le problème en cours. Autrement dit, on n'utilise pas des cas similaires puisqu'ils sont généralisés par le réseau de neurones qui les représente. Il n'y a pas une interaction directe entre la fonction d'adaptation et la base de cas (Figure 5-3). Les connaissances d'adaptation sont représentées par les poids des liens (arcs) des réseaux de neurones, qui sont ajustés au cours de la phase d'apprentissage des réseaux. Par conséquent, notre technique d'adaptation n'utilise que les poids des arcs comme connaissances d'adaptation. Ces dernières sont générées à partir de la base de cas.
- Pendant la phase de révision, si l'utilisateur juge que la solution proposée par

le système RÀPC n'est pas pertinente, et après qu'il ajoute des modifications, nous proposons de faire un nouvel apprentissage du réseau qui l'a fournie par le nouveau cas révisé par l'utilisateur. Ce même cas sera présenté au réseau plusieurs fois jusqu'à ce que l'erreur soit inférieure à un seuil donné, ou bien on va atteindre un nombre d'itérations fixé par l'utilisateur ou le concepteur du système. Cette phase peut être suivie d'une phase d'apprentissage ou de mémorisation, visant à capitaliser le nouveau cas.

3. Cas d'application : adaptation de scénarios pédagogiques

Nous avons testé et utilisé notre technique d'adaptation pour adapter des scénarios pédagogiques aux apprenants modélisés par le modèle IMS LIP. Nous allons tout d'abord, présenter ce modèle de représentation d'apprenants.

3.1. Présentation du modèle IMS LIP

Plusieurs modèles de représentation d'apprenants existent dans la littérature : IMS LIP, PAPI (IEEE LTSC Personal and Private Information Standard), eduPerson (Internet2 et Educause), Dolog LP, FOAF (Friend Of A Friend)...etc. Les plus célèbres entre ces modèles sont le modèle IMS LIP et le modèle PAPI.

IMS-LIP⁴ (Information Management System - Learner Information Package) est un modèle de représentation d'apprenants. Son but est de modéliser les apprenants en structures de données pour ensuite, échanger ces structures entre différents systèmes. Sa première version de spécification 1.0 a été réalisée en mars 2001 et comporte trois documents :

- IMS Learner Information Package Information Model ;
- IMS Learner Information Package XML Binding;
- IMS Learner Information Packaging Best Practice & Implementation Guide.

Dans cette version (1.0), 11 catégories d'information pour représenter les apprenants sont définies : Identification, Goal, QCL (Qualification-Certification-Licenses), Activity, Interest, Relationships, Competency, Accessibility, Transcript, Affiliation, SecurityKey. Ces catégories sont similaires à ceux trouvées dans un CV (curriculum vitae). Nous les décrivons ci-dessous :

1. *Identification* : Cette catégorie d'informations contient les informations biographiques et démographiques qui permettent d'identifier un apprenant, tels que son nom, son adresse, son E-mail...etc.

⁴ <http://www.imsglobal.org/profiles/index.html>

Champ	Définition	Type
ID_PERSON	Identificateur de l'apprenant	Chaîne de caractères
FIRST_NAME	Prénom de l'apprenant	Chaîne de caractères
FAMILY_NAME	Nom de l'apprenant	Chaîne de caractères
PHOTO	La photo de l'apprenant	Chaîne de caractères
HOME_PAGE	--	Chaîne de caractères
ADDRESS	Adresse personnel de l'apprenant	Chaîne de caractères
EMAIL	Adresse électronique de l'apprenant	Chaîne de caractères
GENDER	--	Concept (Gender)
DT_BIRTH	Date de naissance de l'apprenant	Date
COMMENT	--	Chaîne de caractères
PHONE_FAX	Numéro de téléphone de l'apprenant	Numérique

Tableau 5-1 : Catégorie *Identification* (adapté de [Razmerita, 2003]).

2. *Goal* : Cette catégorie d'informations contient les données sur les objectifs et les sous objectifs personnel de l'apprenant

Attribute	Type
GOAL	Concept (Goal)

Tableau 5-2 : Catégorie *Goal* [Razmerita, 2003].

3. *Qualification, Certification, Licenses (QCL)*: Cette catégorie d'informations regroupe les qualifications, les certifications et les diplômes attribués à un apprenant.

Attribute	Type
COURSES_ATTENDED	Concept (Course)
DEGREES	Concept (Degree)

Tableau 5-3 : Catégorie *QCL* [Razmerita, 2003].

4. *Activity* : Cette catégorie d'informations contient les données sur les activités liées au travail et à la formation d'un apprenant.

Attribute	Type
CONFERENCES	Chaîne de caractères
WORKING PAPERS	Chaîne de caractères
PROJECTS	Concept (Project)
WHITE_PAPERS	Chaîne de caractères
DOCUMENTATIONS	Chaîne de caractères
PROTOTYPES	Chaîne de caractères

Tableau 5-4 : Catégorie *Activity* (adapté de [Razmerita, 2003]).

5. *Interest* : Cette catégorie d'informations regroupe les données sur les

hobbies et les activités créatrices d'un apprenant.

Attribute	type
HOBBIES	Chaîne de caractères

Tableau 5-5 : Catégorie *Interst* (adapté de [Razmerita, 2003]).

6. *Relationships* : Cette catégorie d'informations permet de décrire les relations entre les structures de données utilisées pour stocker les données de la personne employée dans ce modèle.

7. *Competency* : Cette catégorie d'informations décrit les compétences associées avec la formation formelle ou informelle d'une personne et son expérience de travail.

Attribute	type
COMPETENCIES	Concept (Competencies)
C_LEVEL	Chaîne de caractères
EXPERTISE_FIELDS	Chaîne de caractères
E_LEVEL	Chaîne de caractères

Tableau 5-6 : Catégorie *Competency* (adapté de [Razmerita, 2003]).

8. *Accessibility* : Cette catégorie d'informations permet de décrire les préférences de la personne, ses langues, et ses éventuels handicaps.

Attribute	Type
LANGUAGES	Concept (Languages)
INTERST_AREAS	Concept (Interest_Areas)
USER PREFERENCES	Concept (Media type)
MOOD	Chaîne de caractères
SOCIAL NETWORK	Concept (UserRelationship)
ROLE	Chaîne de caractères
LEARNING STYLE	Concept (Learning_Style)
COGNITIVE STYLE	Concept (Cognitive_Style)

Tableau 5-7 : Catégorie *Accessibility* [Razmerita, 2003].

9. *Transcript* : Cette catégorie d'informations permet de décrire les données sur les bulletins de notes de l'apprenant.

10. *Affiliation* : Cette catégorie d'informations inclut les informations sur la description de l'organisation associée à l'apprenant.

Champ	Type
TITLE	Concept (Title)
WORK_UNIT	Chaîne de caractères
TEAM_NAME	Chaîne de caractères
SUPERVISOR	Chaîne de caractères
START_DATE	Date
PEOPLE_REPORTING	Chaîne de caractères

Tableau 5- 8 : Catégorie *Affiliation* (adapté de [Razmerita, 2003]).

11. *SecurityKey* : Cette catégorie d'informations regroupe les données de sécurité d'un apprenant, tels que les mots de passe et les codes de sécurité qui doivent être utilisés durant ses communications. Pour chaque clé de sécurité une structure différente sera utilisée.

Champ	Type
User_Name_1	Chaîne de caractères
Password_1	Chaîne de caractères
User_Name_2	Chaîne de caractères
Password_2	Chaîne de caractères

Tableau 5- 9 : Catégorie *SecurityKey* (adapté de [Razmerita, 2003]).

La Figure suivante illustre ces 11 catégories qui constituent le modèle IMS LIP:

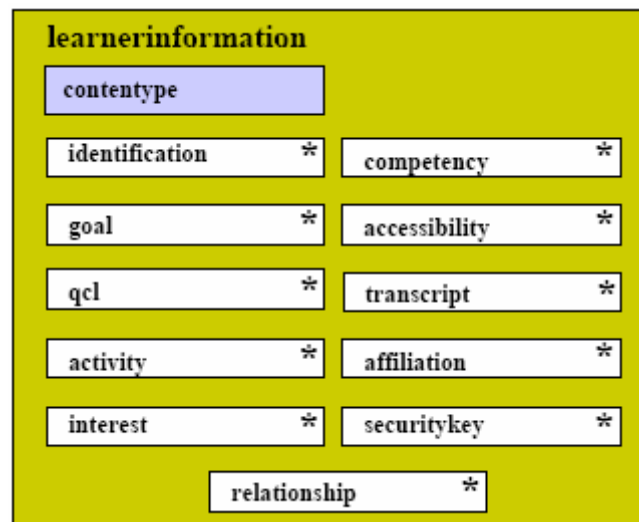


Figure 5- 4 : Catégories d'IMS LIP [IMS].

3.2. Adaptation des scénarios d'apprentissage par la technique proposée

Nous avons testé notre technique (décrite dans la section 2) en ingénierie pédagogique, plus précisément dans l'adaptation de scénarios d'apprentissages aux apprenants. Pour cela, nous avons modélisé cette problématique en terme de RàPC. Le problème représente quelques caractéristiques de l'apprenant selon le modèle IMS LIP (présenté dans la section 3.1), et la solution représente le scénario d'apprentissage qui lui soit approprié, dans le cadre d'un dispositif de formation en ligne.

L'idée est de capitaliser les anciens cas, à savoir les anciens apprenants, pour lesquels les scénarios ont été déjà choisis par leurs tuteurs. Nous considérons ainsi que ces scénarios constituent des solutions sources à ces apprenants (problèmes sources).

Il s'agit ensuite de rechercher, lors de l'arrivée d'un nouvel apprenant, le cas pivot le plus proche parmi les cas pivot qui existent. Ensuite, nous présentons les caractéristiques du nouvel apprenant à l'entrée du réseau associé au GDS du cas pivot le plus proche. Après calcul, le réseau de neurones fournira un scénario adapté à cet apprenant.

Nous avons choisi d'adapter les scénarios d'apprentissage pour le module « System Engineering » assuré par Mr CHIKH Azeddine, le directeur de ce mémoire, dans différents postes de graduation. Pour cela, nous avons pris un exemple d'un cours de ce module qui est formé de 6 activités différentes auxquelles nous ajoutons une activité nommée « Null » pour uniformiser la taille des scénarios (vecteur de sortie).

Exemple :

Le premier tableau ci-dessous illustre quelques exemples de profils d'apprenants que nous avons utilisés pour décrire les apprenants en utilisant quelques caractéristiques du modèle IMS-LIP.

id	Age	Ville	Sexe	Magistère	Moyenne-concours	Moyenne-Cursus	Rupture
Learner 1	26	Tlemcen	Homme	SRT	12	15	Non
Learner 2	30	Tlemcen	Femme	API	14.5	12.25	Oui
Learner 3	24	Tiaret	Homme	STIC	16	13.75	Non
Learner 4	28	Tlemcen	Femme	SRT	13.75	14.5	Oui
Learner 5	24	Tiaret	Homme	STIC	16	13.75	Non
Learner 6	24	Tlemcen	Homme	API	12.5	13	Non
Learner 7	28	Tiaret	Homme	STIC	14.8	15.75	Oui
Learner 8	24	Tlemcen	Femme	API	14	13.25	Non
Learner 9	27	Tiaret	Homme	STIC	12.5	12.5	Oui
Learner 10	24	Tlemcen	Femme	API	14	13.5	Non

Tableau 5-10 : Exemples d'apprenants.

Nous avons supposé que :

- L'âge d'un apprenant est compris entre 20 et 35 ans ;
- La ville de l'apprenant peut prendre deux valeurs possibles (Tlemcen, Tiaret) ;
- La caractéristique Magistère peut prendre 3 valeurs possibles (API, SRT, STIC);
- L'attribut rupture est utilisé pour indiquer si l'apprenant a fait une rupture entre ses études de graduation et les études de post-graduation.

Le second tableau ci-dessous illustre quelques exemples de scénarios pédagogiques, représentés sous forme de suites ordonnées d'activités dans un dispositif de formation en ligne.

	1	2	3	4	5	6
Scenario1	A2	A3	A1	A4	A5	Null
Scenario2	A1	A4	A3	A5	A2	A6
Scenario3	A2	A1	A5	A4	Null	Null
Scenario4	A2	A1	A5	A4	A3	Null
Scenario5	A3	A1	A5	A4	Null	Null
Scenario6	A2	A1	A5	A4	A3	A6
Scenario7	A3	A1	A5	Null	Null	Null
Scenario8	A3	A1	A5	A4	A6	A2
Scenario9	A2	A1	A5	A4	A6	Null
Scenario10	A6	A1	A5	A4	A3	A2

Tableau 5- 11 : Exemples de scénarios d'apprentissage.

Une activité peut être un document pdf, une présentation power point, une vidéo, une page html, un chapitre d'un livre...etc.

3.2.1. Saisie des cas

La première étape dans les systèmes de RàPC est de saisir les exemples (cas) déjà établis (les apprenants pour lesquels les scénarios sont déjà connus et définis par leur tuteur). Si la base de cas est vide, le système de RàPC ne peut pas proposer une solution à un problème donné.

The 'Saisie des cas' window contains the following fields and values:

- Code: 1
- Age: 29
- Ville: Tlemcen
- Sexe: H
- Moy_concourt: 12
- Moy_cursus: 14
- Rupture: 0
- Nom_Magi: SRT
- Code_scenario: 8

Figure 5- 5 : la fenêtre de saisie.

3.2.2. Création des GDS

The 'Création des GDS' window includes the following filters and a data table:

- Nom_Magi: SRT
- Rupture: 0
- Sexe: H
- Moy_concourt entre: [] et []
- Moy_cursus: [] et []
- Age Entre: [] et []
- Ville: []
- Créer ce GDS button

Code	Age	Ville	Sexe	Moy_concourt	Moy_cursus	Rupture	code_sc
1	29	Tlemcen	H	12	14	0	
4	30	Tiaret	H	15	16	0	
10	24	Tlemcen	H	17	18,5	0	

Figure 5- 6 : Fenêtre pour la création de GDS.

Nous avons utilisé les caractéristiques (Magistère, Sexe, Rupture) pour former les GDS. Pour cela, nous avons obtenu 12 GDS :

GDS1 : regroupe les apprenants dont (Magistère=API, Sexe=H, Rupture=Oui) ;

GDS2 : regroupe les apprenants dont (Magistère=API, Sexe=H, Rupture=Non) ;
 GDS3 : regroupe les apprenants dont (Magistère=API, Sexe=F, Rupture=Oui) ;
 GDS4 : regroupe les apprenants dont (Magistère=API, Sexe=F, Rupture=Non) ;
 GDS5 : regroupe les apprenants dont (Magistère=SRT, Sexe=H, Rupture=Oui) ;
 GDS6 : regroupe les apprenants dont (Magistère=SRT, Sexe=H, Rupture=Non) ;
 GDS7 : regroupe les apprenants dont (Magistère=SRT, Sexe=F, Rupture=Oui) ;
 GDS8 : regroupe les apprenants dont (Magistère=SRT, Sexe=F, Rupture=Non) ;
 GDS9 : regroupe les apprenants dont (Magistère=STIC, Sexe=H, Rupture=Oui) ;
 GDS10 : regroupe les apprenants dont (Magistère=STIC, Sexe=H, Rupture=Non) ;
 GDS11 : regroupe les apprenants dont (Magistère=STIC, Sexe=F, Rupture=Oui) ;
 GDS12 : regroupe les apprenants dont (Magistère=STIC, Sexe=F, Rupture=Non) ;

Cependant, l'utilisateur peut créer des GDS en utilisant les autres caractéristiques des apprenants.

3.2.3. Sélection de cas pivots

Pour la sélection de cas pivots, nous avons calculé pour chacun des attributs (Age, Moyenne-concours et Moyenne-cursus) la moyenne de tous les cas qui constituent chaque GDS. Pour l'attribut Ville, nous avons compté le nombre d'occurrence des deux valeurs possibles, celle qui est la plus fréquente sera retenue pour former le cas pivot.

3.2.4. Paramètres des réseaux de neurones

Pour notre application, nous avons utilisé les paramètres suivants pour chaque réseau de neurones :

- **Codage des entrées (caractéristiques des apprenants)**

Pour la couche d'entrée, nous avons utilisé 7 neurones, un pour chaque caractéristique d'apprenant.

Nous avons codé les entrées du réseau de la manière suivante :

- Les attributs Age, Moyenne-concours et Moyenne-cursus sont présentés à l'entrée du réseau comme ils sont (pas de changement) ;
- L'attribut Ville est représenté par 1 pour la valeur 'Tlemcen' et 0 pour 'Tiaret' ;
- Même chose pour les attributs Sexe (1 pour 'H' et 0 pour 'F') et Rupture (1 pour 'Oui' et 0 pour 'Non') ;
- L'attribut magistère est codé par : 0 pour API, 1 pour SRT et 2 pour STIC.

- **Codage des sorties (scénarios d'apprentissage)**

Pour la couche de sortie, nous avons utilisé 12 neurones pour distinguer quel scénario est proposé à un apprenant.

Chaque scénario est formé de 6 activités dont au moins 2 non Null. La première activité peut prendre 6 valeurs possibles (A1 → A6). La deuxième peut prendre 5 valeurs...etc. Donc, nous aurons $6 \times 5 \times 5 \times 4 \times 3 \times 2 = 3600$ scénarios possibles formés des 7 activités possibles ; ce qui nécessite 12 bits

pour coder tous les scénarios sous la forme binaire ($2^{12} = 4096$ supérieur à 3600).

- **Couches cachets**

Nous avons utilisé 2 couche cachées pour les réseaux de neurones, la première comporte 9 neurones et la deuxième 11 neurones. Ce choix est heuristique. Donc, nos réseaux de neurones comporte respectivement 7, 9, 11, 12 neurones pour les couches 1 (d'entrée), 2, 3 et 4 (de sortie).

3.2.5. Processus de raisonnement

A l'arriver d'un nouvel apprenant, le système RàPC lui demande de saisir ses caractéristiques qui va les utilisés pour lui proposé un scénario qui lui convient (Figure 5-7). Ensuite, le système RàPC va comparer ces caractéristiques avec le cas pivot de chaque GDS. Si le système RàPC juge que ce nouveau problème (le nouvel apprenant) est similaire à un cas pivot alors, le réseau de neurones associé au GDS de ce cas pivot sera utilisé par le système RàPC pour générer une nouvelle solution (scénario) à ce nouvel apprenant.

Si le scénario proposé par le système RàPC est jugé pertinent, il sera automatiquement mémorisé dans la base de cas pou qu'il sera réutilisé dans les future raisonnements, sinon, après que l'utilisateur ajoute des modification sur la solution proposée par le système, ce dernier va ajouter cette solution révisée à sa base de cas et

Code	Age	Ville	Sexe	Moy_concours	Moy_cursus	Rupture	Code_s

Figure 5- 7 : fenêtre d'adaptation.

un nouveau apprentissage du réseau de neurones qui a fourni cette solution sera lancé en utilisant seulement ce cas révisé plusieurs fois.

4. Comparaison de la technique proposée avec celle de [Corchado 2001]

Nous avons proposé une technique d'adaptation et d'ACA basée sur l'utilisation des réseaux de neurones de type PMC. Cette technique est plus ou moins facile à mettre en œuvre par rapport à d'autres techniques. C'est une technique d'adaptation automatique et dérivationnelle. Elle est automatique parce qu'elle fait l'ACA à partir de la base de cas. Elle est dérivationnelle parce qu'elle utilise le processus qui permet de passer d'un problème à sa solution.

Cette technique ressemble à celle de [Chorchado et al., 2001] puisqu'elles utilisent toutes les deux les réseaux de neurones pour effectuer l'adaptation.

Les apports de notre technique par rapport à l'autre sont :

1. L'apprentissage des réseaux de neurones se fait avant le processus de raisonnement, ce qui permet un gain de temps important.
2. Les réseaux entraînés seront sauvegardés pour des futurs raisonnements.
3. Le fait de fragmenter la base de cas en GDS augmente l'efficacité du système et les résultats fournis seront plus adéquats.
4. La sélection d'un cas pivot pour chaque GDS permet de réduire énormément l'effort de remémoration.
5. Le fait d'utiliser les réseaux de neurones PMC augmente la généralisation puisque ces derniers donnent de bons résultats par rapport aux réseaux RBF dans la résolution de problèmes complexes. Le seul point faible des PMC par rapport aux RBF est leur apprentissage qui est un peu long vis-à-vis des RBF qui apprennent beaucoup plus vite. Cependant, cette contrainte ne va pas influencer sur le fonctionnement du système RÀPC puisque les réseaux de neurones PMC seront préparés à l'avance et donc, l'utilisateur ne sera pas pénalisé.

5. Conclusion

Dans ce chapitre, nous avons présenté une nouvelle technique d'adaptation et d'acquisition de connaissances d'adaptation qui consiste à utiliser les réseaux de neurones de type PMC pour résoudre de nouveaux problèmes. Un cas d'application est aussi présenté qui est l'adaptation des scénarios d'apprentissage à des apprenants modélisés par le modèle IMS LIP.

Cette nouvelle technique est appliquée dans ce domaine et les résultats obtenus sont jugés acceptables.

Conclusion générale

Nous avons présenté dans ce travail de recherche une nouvelle technique d'adaptation des cas dans les système de RàPC. Cette technique utilise les réseaux de neurones de type PMC pour effectuer la tâche de l'adaptation. Elle est caractérisée par le fait qu'elle partage la base de cas en groupe de cas similaire, elle sélectionne un cas pivots pour chaque groupe et le faite qu'elle associée à chaque groupe un réseau de neurones qui sera entraîné en utilisant les cas du groupe associé.

Cette technique présente quelques avantages tels que : l'acquisition de connaissances d'adaptation est automatique et simple, les réseaux de neurones seront préparés (entraînés) avant le processus de raisonnement ce qui ne va pas perturber l'utilisateur du système RàPC.

L'inconvénient majeur de cette technique est celui des réseaux de neurones (la taille du réseau et surtout de celle de la couche de sortie). Si cette dernière contient peu de neurones, ne nous pouvons pas coder beaucoup d'exemples. Par contre si ce nombre est élevé, la complexité du réseau augmente et le temps de calcul sera plus considérable.

Ce projet de recherche nous a permis de découvrir le monde de l'intelligence artificielle et ces techniques les plus utilisées notamment le raisonnement à partir de cas et les réseaux de neurones.

Comme perspectives, nous proposons de tester cette technique dans d'autres domaines d'application pour évaluer sa pertinence et pour mieux juger son utilité. Nous proposons aussi d'utiliser une technique de classification automatique tel que les réseaux de neurones pour générer les GDS.

Une technique d'adaptation des cas basée sur les réseaux de neurones

Réda Bouabdallah¹, Azeddine Chikh²

Département d'informatique - Faculté des sciences de l'ingénieur

Université de Tlemcen - Algérie

¹redainfosrt@yahoo.fr, ²az_chikh@univ-tlemcen.dz

Résumé

Le raisonnement à partir de cas (RàPC) est un paradigme de l'intelligence artificielle. Il consiste à résoudre un nouveau problème en se remémorant et en adaptant un ou plusieurs cas passés déjà résolus. L'adaptation constitue l'étape la plus importante du processus de RàPC. Elle nécessite l'acquisition d'un ensemble de connaissances spécifiques généralement très difficiles à modéliser. Dans cet article, nous proposons une technique d'adaptation des cas par l'utilisation de réseau de neurones. On fournit à l'entrée du réseau le nouveau problème rencontré et on récupère à sa sortie la solution adaptée. Cette technique présente deux principaux avantages : (1) l'apprentissage des réseaux de neurones se fait avant le processus de raisonnement ; (2) les réseaux générés peuvent être réutilisés dans les futures adaptations. Elle sera appliquée à l'adaptation des scénarios d'apprentissage pédagogique, décrits sous forme de suites d'activités, aux apprenants décrits avec le modèle IMS-LIP.

Mots clés : Raisonnement à partir de cas (RàPC), adaptation, acquisition de connaissances d'adaptation (ACA), réseaux de neurones, scénario d'apprentissage, apprenant.

1 Introduction

Le raisonnement à partir de cas (RàPC) est inspiré des travaux sur la mémoire dynamique [10]. C'est un cas particulier du raisonnement par analogie. Il est basé sur l'hypothèse que « des problèmes similaires ont des solutions similaires ». Le RàPC est une approche de résolution de problèmes basée sur la réutilisation d'expériences passées appelées "cas sources" pour la résolution de nouveaux problèmes appelés "problèmes cibles". Un cas est décomposé en deux parties : la première décrit le problème et la seconde sa solution. Chaque partie est généralement représentée par un groupe de descripteurs où chaque descripteur est formé d'un couple (attribut, valeur). L'ensemble des cas constitue la base de cas.

Dans cet article, nous présentons une technique d'adaptation des cas et d'acquisition de connaissances d'adaptation fondée sur l'utilisation des réseaux de neurones. Nous décomposons d'abord la base de cas en groupes de similarités (GDS), où chaque groupe rassemble un ensemble de cas qui partagent un même sous ensemble d'attributs. Nous associons ensuite, à chaque groupe un réseau de neurones qui sera entraîné en utilisant les cas de ce groupe. Dans la phase d'adaptation du cycle de RàPC, il ne nous reste qu'à présenter le nouveau problème à l'entrée du réseau de neurones associé au problème en cours. La solution du nouveau problème, récupérée à la sortie du réseau est obtenue après un calcul effectué par ce dernier.

Cette introduction est suivie d'une description de la phase de l'adaptation en RàPC (section 2). La section 3 présente un état de l'art des techniques d'adaptation et d'acquisition des connaissances d'adaptation. Dans la section 4, nous présentons notre contribution, qui consiste à proposer une nouvelle technique d'adaptation et d'acquisition des connaissances d'adaptation utilisant les réseaux de neurones. Dans la section 5, nous montrons un cas d'application relatif à l'adaptation des scénarios

pédagogiques aux apprenants. Enfin, nous montrons en conclusion, l'apport de cette nouvelle technique.

2 L'adaptation en RàPC

Le processus de RàPC comporte 5 étapes séquentielles qui sont : (1) l'élaboration ; (2) la remémoration ; (3) l'adaptation ; (4) la révision et (5) l'apprentissage.

L'adaptation est l'une des étapes les plus difficiles du RàPC [6]. Elle permet d'obtenir une solution au problème cible à partir de la solution du cas source sélectionné au cours de la phase de remémoration. Cette dernière est éventuellement adaptée afin de satisfaire les contraintes du problème posé et de tenir compte des différences entre les problèmes source et cible.

L'adaptation s'appuie sur des connaissances d'adaptation qui peuvent prendre différentes formes (règles d'adaptation, cas d'adaptation...). Ces connaissances sont généralement de nature implicite et leur explicitation reste une tâche difficile.

Dans cette étape, la majorité des systèmes RàPC sollicitent l'aide des utilisateurs pour participer à la création d'une nouvelle solution à partir de la solution remémorée. Ceci est dû à la difficulté de l'implémentation de cette phase [2].

Le but de l'adaptation est donc de proposer une solution au problème cible en modifiant la solution de problème source. Donc, on assiste à la présence d'un système de raisonnement par analogie :

« Sachant que *la solution du problème cible est à problème cible ce que la solution du problème source est à problème source*, connaissant *problème source, solution du problème source et problème cible*, que vaut *la solution du problème cible ?* ».

Le carré d'analogie (Figure 2) peut être lu autrement : « Sachant que *la solution du problème cible est à la solution du problème source ce que le problème cible est au problème source*, connaissant *le problème source, la solution du problème source et le problème cible*, que vaut *la solution du problème cible ?* » (Adaptée de [3; 4]).

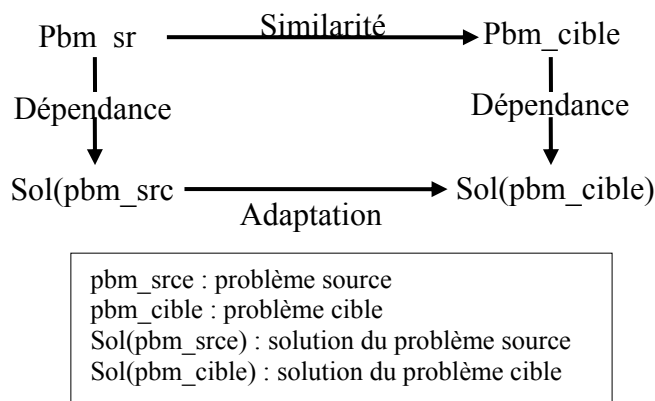


Fig. 1. Schéma d'adaptation.

Il existe deux dimensions d'adaptation [4] :

- **La dimension « produit » (transformationnelle, structurelle)**

La solution trouvée n'est pas une solution directe pour le nouveau problème, mais il existe une connaissance sous forme des opérateurs transformationnels qui sont appliqués à la solution ancienne. On distingue deux sous classes dans l'adaptation transformationnelle :

- *Par valeur* : elle consiste à adapter les valeurs des attributs de la solution

du cas source.

- *Par structure* : elle consiste à adapter la structure de la solution du cas source.

- **La dimension « processus » (dérivationnelle, générative)**

Elle consulte les informations existant dans le cas remémoré. Ces informations contiennent la méthode utilisée pendant la résolution de l'ancien problème. Il s'agit donc, d'exploiter la démarche de résolution utilisée dans la résolution de l'ancien problème pour créer une nouvelle solution qui correspond au nouveau problème. Dans cette classe d'adaptation, on garde pour chaque cas passé une trace des étapes qui ont permis de générer sa solution. Cette traçabilité est souvent utilisée en logique de conception (Design rationale).

3 Techniques d'adaptation et d'acquisition de connaissances d'adaptation

Beaucoup de travaux se sont intéressés à la phase d'adaptation et à l'acquisition des connaissances associées. Un état de l'art détaillé et une étude comparative de travaux sur l'adaptation et l'ACA sont présentés dans [8]. Cependant, nous ajoutons d'autres techniques qui ne sont pas citées dans cette étude et qui ont un rapport avec notre technique :

[5] propose une technique qui utilise un réseau de neurones artificiels de type RBF (radial basis function) pour réaliser la tâche d'adaptation : l'apprentissage du réseau est réalisé par un ensemble de cas sources similaires aux nouveau problème, sélectionnés lors de la phase de remémoration (par la méthode des k plus proches voisins). Le problème cible sera représenté à l'entrée du réseau et sa solution est récupérée à sa sortie. Cette technique effectue bien l'acquisition de connaissances d'adaptation. La fonction d'adaptation est bien maîtrisée par le réseau de neurones grâce aux cas sources sélectionnés dans la phase de remémoration et qui sont utilisés dans l'apprentissage du réseau. Le problème majeur de cette technique est qu'après chaque résolution, aucune connaissance d'adaptation ne sera retenue par le système. Autrement dit, un réseau de neurones ne sert qu'à résoudre un problème donné et il sera rejeté. Le domaine d'application de cette technique est la prévision de la température de l'océan à une profondeur fixe, pour les prochains kilomètres parcourus par un navire.

Dans [11, 12], les auteurs proposent une technique de remémoration guidée par l'adaptation. Cette technique suggère d'utiliser les connaissances d'adaptation pendant la phase de remémoration. Le cas sélectionné est celui qui réduit l'effort de l'adaptation et qui résout mieux le nouveau problème. Cette technique est appliquée pour la conception de logiciels de pilotage industriels.

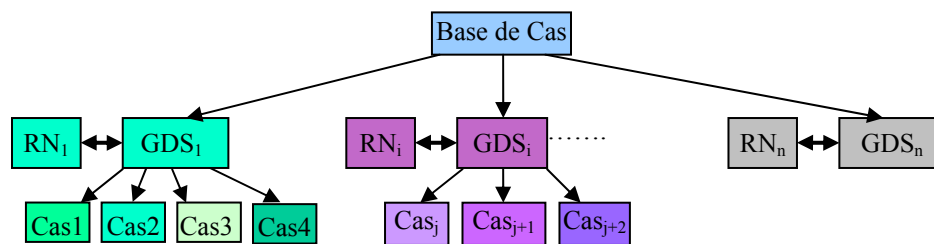
[1] propose une technique qui utilise les algorithmes génétiques pour effectuer l'adaptation des cas. Les cas sélectionnés lors de la phase de remémoration forment la population initiale pour les algorithmes génétiques. Deux types d'adaptation sont utilisés pour faire évoluer le contenu des cas remémorés jusqu'à obtenir une solution satisfaisant les contraintes du domaine d'application : (1) structurelle : par combinaison ou croisement, (2) paramétrique : par mutation. Elle est appliquée au

système GENCAD pour la conception de tours et la disposition des résidences selon l'art de conception chinois « fong shui ».

4 Technique proposée

Notre technique d'adaptation est basée sur les réseaux de neurones. Il s'agit de décomposer d'abord la base de cas en Groupes De Similarité (GDS). Nous associons ensuite à chaque groupe un réseau de neurone qui va être entraîné dans la phase d'apprentissage en utilisant les cas du groupe associé. Nous aurons donc autant de réseaux de neurones qu'il y'a de GDS.

La Figure 2 ci-dessous illustre l'organisation arborescente de notre base de cas.



GDS: Groupe De Similarité
RN : Réseau de Neurones

Fig. 2. Décomposition de la base de cas.

4.1 Création des GDS

Pour la création des GDS, nous proposons d'utiliser un sous groupe d'attributs de la partie problème, jugés les plus discriminants. Le choix de ces attributs va influencer la qualité du système, d'où la nécessité de les sélectionner soigneusement pour aboutir à des résultats pertinents. Les cas d'un même GDS ont des valeurs égales pour le sous groupe d'attributs utilisé pour la discrimination.

4.2 Sélection des cas pivots

Après la création des GDS, nous suggérons de sélectionner un cas pivot pour chaque GDS pour réduire l'effort de la remémoration. Ainsi, au lieu de mesurer la similarité du nouveau problème avec tous les cas de la base, nous le comparons seulement avec les cas pivots. Nous proposons deux façons pour la sélection des cas pivots :

- Soit d'une manière automatique par le système de RàPC en calculant la moyenne de chaque attribut des cas d'un GDS. Le cas le plus proche du vecteur formé des moyennes calculées pour chaque attribut sera choisi comme cas pivot ;
- Soit manuellement en laissant le choix à l'utilisateur.

L'idée d'utiliser des cas pivots vient du constat que dans notre cas, si un problème est similaire à deux cas qui appartiennent au même GDS, le résultat de

l'adaptation sera la même puisque les deux cas ont un même réseau de neurone qui va faire l'adaptation.

4.3 Apprentissage des réseaux de neurones

Après avoir partagé la base de cas en groupes de similarité, nous allons créer un réseau de neurone de type PMC (Perceptron Multi-couches) pour chacun de ces groupes. Ensuite, un apprentissage supervisé de chacun de ces réseaux, par les cas que contient le groupe associé, est réalisé. Le résultat obtenu par le réseau va être comparé avec la partie solution de ce même problème. En calculant l'écart entre eux, ce dernier permettra de modifier les poids des arcs du réseau. Ce processus est répété jusqu'à ce que l'écart soit inférieur à un seuil fixé par l'utilisateur ou bien le nombre d'itérations atteint une certaine limite fixée par l'utilisateur.

La Figure 3 ci-dessous illustre le déroulement du processus décrit précédemment sur un exemple. Etant donné un cas source : (p_{bm_sorce}, Sol(p_{bm_sorce})), et que p_{bm_sorce} contient les attributs A1, A2 et A3 et la Sol(p_{bm_sorce}) les attributs A4, A5 et A6. Les attributs de la partie p_{bm_sorce} (A1, A2, A3) seront présentés à l'entrée du réseau de neurones associé au GDS affecté au problème en cours. Après calcul (propagation), le réseau de neurones va fournir une solution (A7, A8, A9). En comparant cette dernière avec la solution réelle [Sol(p_{bm_sorce})=(A4, A5, A6)], on obtient une différence $\Delta Sol=(A4-A7, A5-A8, A6-A9)$. Les poids des arcs du réseau de neurones seront raffinés à partir de ΔSol et ajustés par la technique de rétro-propagation du gradient en vue d'optimiser l'adaptation.

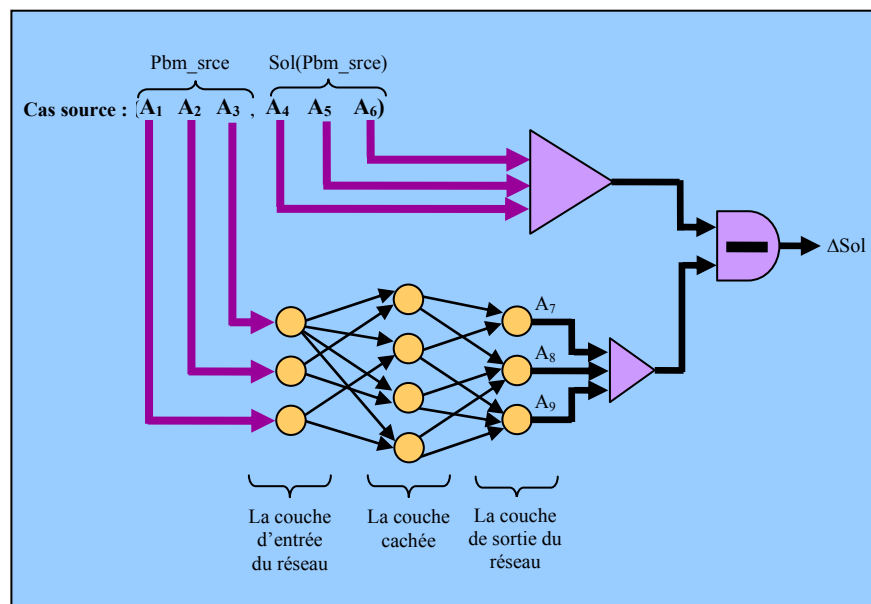


Fig. 3. Apprentissage des réseaux de neurones.

Après cette étape, les réseaux seront stockés dans une base de données et seront prêt à être utilisés par le système de RàPC.

4.4 Processus du raisonnement

Nous décrirons ci-dessous les trois principales phases qui se trouvent au coeur du processus RàPC :

- La phase de remémoration, qui vient après la phase d'élaboration, permet d'affecter le nouveau problème au GDS du cas pivot le plus proche. Si la distance entre le nouveau problème et le cas pivot le plus proche est supérieure à un seuil donné, le nouveau problème sera rangé, s'il est jugé pertinent, dans un groupe spécifique (dont les cas qui le composent ne sont pas forcément similaires). Ce groupe spécifique sera utilisé ultérieurement comme nouvelle base qui sera décomposée à son tour en GDS. L'idée d'avoir un GDS qui inclut des cas non similaires ressemble à l'idée d'avoir des zones de cas atypique proposé par Malek [9].

- Pendant la phase d'adaptation, le nouveau problème sera présenté alors à l'entrée du réseau de neurones associé au GDS du cas pivot le plus proche. Après calcul, le réseau fournira en sortie la solution adaptée (Figure 4). Dans notre cas, la fonction d'adaptation qui se résume au réseau de neurones, n'utilise comme entrée que le problème en cours. Autrement dit, on n'utilise pas des cas similaires puisqu'ils sont généralisés par le réseau de neurones qui les représente. Il n'y a pas une interaction directe entre la fonction d'adaptation et la base de cas (Figure 4). Les connaissances d'adaptation sont représentées par les poids des liens (arcs) des réseaux de neurones, qui sont ajustés au cours de la phase d'apprentissage des réseaux. Par conséquent, notre technique d'adaptation n'utilise que ces poids comme connaissances d'adaptation. Ces dernières sont raffinées en utilisant la base de cas.

- Pendant la phase de révision, si l'utilisateur juge que la solution proposée par le système de RàPC n'est pas pertinente, et après qu'il ajoute des modifications, nous proposons de faire un nouvel apprentissage du réseau, qui a fournie la solution refusée, par le nouveau cas révisé par l'utilisateur. Ce même cas sera présenté au réseau plusieurs fois jusqu'à ce que l'erreur soit inférieure à un seuil donné, ou bien on va atteindre un nombre d'itérations fixé par l'utilisateur ou le concepteur du système. Cette phase peut être suivie d'une phase d'apprentissage ou de mémorisation, visant à capitaliser le nouveau cas.

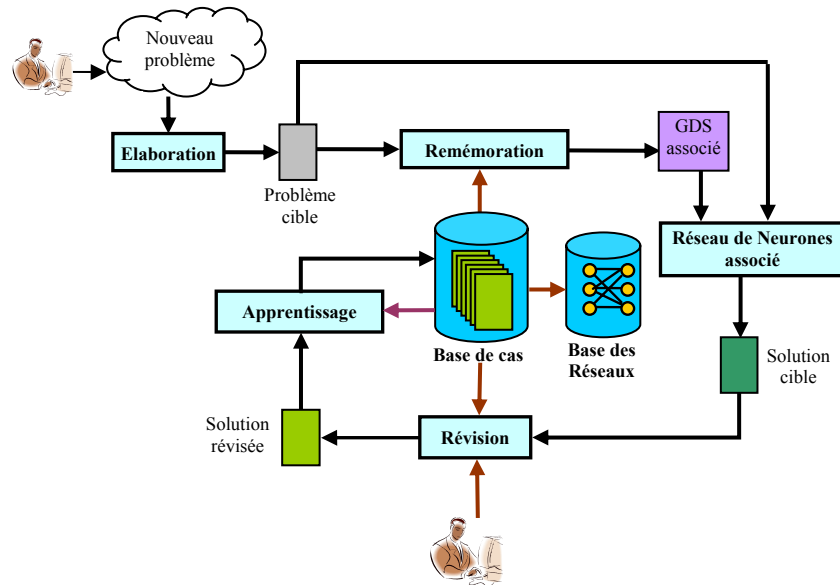


Fig. 4. Notre processus de RàPC.

5 Cas d'application

Nous avons utilisé cette technique d'adaptation en ingénierie pédagogique plus précisément dans l'adaptation à l'apprenant. Le problème représente l'apprenant et la solution représente le scénario d'apprentissage, dans le cadre d'un dispositif de formation en ligne, qui lui soit approprié. L'idée est de capitaliser les anciens cas, à savoir les anciens apprenants, pour lesquels les scénarios ont déjà été choisis par leurs tuteurs. Nous considérons ainsi que ces scénarios constituent des solutions sources à ces apprenants (problèmes sources).

Il s'agit ensuite de rechercher, lors de l'arrivée d'une nouvel apprenant, l'apprenant le plus proche parmi les anciens apprenants. Le scénario utilisé par ce dernier sera alors adapté au nouvel apprenant.

Nous avons choisi de faire la modélisation des apprenants (problème) avec le modèle IMS-LIP [14] et celle des scénarios pédagogiques (solution) à travers un chemin de précedence de leurs activités. Le problème sera ainsi défini comme un vecteur de taille fixe de caractéristiques pertinentes choisies parmi celles du modèle IMS-LIP. La solution sera définie comme un vecteur de taille fixe décrivant l'ordre de déroulement des activités dans chaque scénario.

Nous avons choisi d'adapter les scénarios d'apprentissage pour le module « System Engineering » assuré en post graduation. Pour cela, nous avons pris un exemple d'un cours de ce module qui est formé de 6 activités différentes.

– Exemple :

Le premier tableau ci-dessous illustre quelques exemples de profils d'apprenants décrits avec des caractéristiques d'IMS-LIP.

id	Age	Ville	Sexe	Magistère	Moy. Conc.	Moy. cursus	Rupt.
Apprenant 1	26	Tlemcen	Homme	SRT	12	15	Non
Apprenant 2	30	Tlemcen	Femme	API	14.5	12.25	Oui
Apprenant 3	24	Tiaret	Homme	STIC	16	13.75	Non
Apprenant 4	28	Tlemcen	Femme	SRT	13.75	14.5	Oui
Apprenant 5	24	Tiaret	Homme	STIC	16	13.75	Non

Apprenant 6	24	Tlemcen	Homme	API	12.5	13	Non
Apprenant 7	28	Tiaret	Homme	STIC	14.8	15.75	Oui
Apprenant 8	24	Tlemcen	Femme	API	14	13.25	Non
Apprenant 9	27	Tiaret	Homme	STIC	12.5	12.5	Oui
Apprenant 10	24	Tlemcen	Femme	API	14	13.5	Non

Tableau 1. Exemples d'apprenants

Age: âge de l'apprenant ;

Magistère : intitulé du magistère dans lequel l'apprenant est inscrit (trois valeurs possibles) ;

Moy.conc. : la moyenne du concours d'accès au magistère obtenue par l'apprenant ;

Moy.Cursus : la moyenne obtenue par l'apprenant durant son cursus ;

Rupt. : pour indiquer le mode d'accès de l'apprenant au magister (directement depuis la graduation ou après avoir exercé).

Nous avons utilisé les caractéristiques « Magistère », « Sexe » et « Rupture » pour former les GDS. Nous avons obtenu 12 GDS :

- GDS1 : (Magistère= « API », Sexe= H, Rupture= Oui) ;
- GDS2 : (Magistère= « API », Sexe= H, Rupture= Non) ;
- GDS3 : (Magistère= « API », Sexe= F, Rupture= Oui) ;
- GDS4 : (Magistère= « API », Sexe= F, Rupture= Non) ;
- GDS5 : (Magistère= « SRT », Sexe= H, Rupture= Oui) ;
- GDS6 : (Magistère= « SRT », Sexe= H, Rupture= Non) ;
- GDS7 : (Magistère= « SRT », Sexe= F, Rupture= Oui) ;
- GDS8 : (Magistère= « SRT », Sexe= F, Rupture=Non) ;
- GDS9 : (Magistère= « STIC », Sexe= H, Rupture= Oui) ;
- GDS10 : (Magistère= « STIC », Sexe= H, Rupture= Non) ;
- GDS11 : (Magistère= « STIC », Sexe= F, Rupture= Oui) ;
- GDS12 : (Magistère= « STIC », Sexe= F, Rupture= Non).

Le second tableau ci-dessous illustre quelques exemples de scénarios pédagogiques, représentés sous forme de suites ordonnées d'activités dans un dispositif de formation en ligne.

	1	2	3	4	5	6
Scenario1	A2	A3	A1	A4	A5	Null
Scenario2	A1	A4	A3	A5	A2	A6
Scenario3	A2	A1	A5	A4	Null	Null
Scenario4	A2	A1	A5	A4	A3	Null
Scenario5	A3	A1	A5	A4	Null	Null
Scenario6	A2	A1	A5	A4	A3	A6
Scenario7	A3	A1	A5	Null	Null	Null
Scenario8	A3	A1	A5	A4	A6	A2
Scenario9	A2	A1	A5	A4	A6	Null
Scenario10	A6	A1	A5	A4	A3	A2

Tableau 2. Exemples de scénarios (NULL est utilisée pour uniformiser la taille du vecteur de sortie relatif au scénario)

Une activité peut être la lecture d'un document pdf ou d'une présentation power point, la visualisation d'une vidéo, la résolution d'un exercice, la consultation d'une page web ...etc.

6 Conclusion

Nous avons proposé une technique d'adaptation et d'ACA basée sur l'utilisation des réseaux de neurones de type PMC. Cette technique est plus ou moins facile à mettre en œuvre par rapport à d'autres techniques. C'est une technique d'adaptation automatique et dérivationnelle. Elle est automatique parce qu'elle fait l'ACA à partir de la base de cas. L'apprentissage des réseaux de neurones se fait en utilisant les cas de la base. Elle est dérivationnelle parce qu'elle utilise le processus qui permet de passer d'un problème à sa solution. Cette technique ressemble à celle utilisée dans [5] puisqu'elles utilisent toutes les deux les réseaux de neurones pour effectuer l'adaptation. Les apports de notre technique par rapport à l'autre sont :

- L'apprentissage des réseaux de neurones se fait avant le processus de raisonnement, d'où un gain de temps important.
- Les réseaux entraînés sont sauvegardés pour des futurs raisonnements.
- Le fait de fragmenter la base de cas en GDS augmente l'efficacité du système et les résultats fournis seront plus adéquats.
- La sélection d'un cas pivot pour chaque GDS permet de réduire énormément l'effort de mémorisation.
- Le fait d'utiliser les réseaux de neurones PMC augmente la généralisation puisque ces derniers donnent de bons résultats par rapport aux réseaux RBF dans la résolution de problèmes complexes. Le seul point faible des PMC par rapport aux RBF est leur apprentissage qui est un peu long vis-à-vis des RBF.

Cependant, l'application de notre technique est limitée par le nombre de neurones de la couche de sortie. Si ce nombre est très grand, le temps de réponse et la taille mémoire occupée par ces réseaux vont aussi augmenter. Nous avons testé notre technique pour l'adaptation de scénarios d'apprentissage et nous avons jugé les résultats obtenus acceptables.

7 Bibliographie

- [1] A. G. de Silva Garza et M. L. Maher. An evolutionary approach to case adaptation. In : Proceedings of Third International Conference on Case-Based Reasoning ICCBR-99 , Monastery Seeon, Munich, Germany : 162–172, 1999.
- [2] M. d'Aquin, S. Brachais, J. Lieber et A. Napoli. Vers une acquisition automatique de connaissances d'adaptation par examen de la base de cas- une approche fondée sur des techniques d'extraction de connaissances dans des bases de données. 12ème Atelier de Raisonnement à Partir de Cas – RàPC'04, Université Paris Nord, Villetaneuse, France : 41–52, 2004.
- [3] J. G. Carbonell. Learning by analogy : Formulating and generalizing plans from past experience. In Machine Learning, chapter 5, Tioga Press : 137–161, 1983.
- [4] J. Carbonell. Derivational analogy ; A theory of reconstructive problem solving and expertise acquisition. In R.S. Michalski, J.G. Carbonell, T.M. Mitchell (eds.) : Machine Learning - An artificial Intelligence Approach, Vol.II, Morgan Kaufmann : 371–392, 1986.

- [5] J. Corchado et B. Lees. Adaptation of cases for case-based forecasting with neural network support. In S. K. PAL, T. S. DILLON & D. S. YEUNG, Eds., *Soft Computing in Case-Based Reasoning*, chapter 13 : 293–319. Springer, 2001.
- [6] A. Cordier, B. Fuchs et A.Mille. De l'Ingénierie à l'Apprentissage des connaissances d'adaptation en Raisonnement à Partir de Cas. *Journées ingénierie des connaissances - IC'2006*, p. 41–50, Toulouse, 2006.
- [7] J. Koehler. Planning from Second Principles. *Artificial Intelligence*, 87:145–186, 1996.
- [8] J. Lieber, M. d'Aquin, S. Brachais et A. Napoli. Une étude comparative de quelques travaux sur l'acquisition de connaissances d'adaptation pour le raisonnement à partir de cas. In Sylvie Salotti, Rushed Kanawati Et Farida Zehraoui, Ed., *actes 12ème atelier raisonnement à partir de cas, RàPC'04*, 2004.
- [9] M. Malek. Hybrid Approaches for Integrating Neural Networks and Case Based Reasoning: From Loosely Coupled to Tightly Coupled Models. *Soft Computing in Case Based Reasoning*, Londres, UK, Springer-Verlag : 73-94. 2000.
- [10] R. Schank. *Dynamic memory ; a theory of reminding and learning in computers and people*. Cambridge University Press, 1982.
- [11] B. Smyth. et M. T. Keane. Using adaptation knowledge to retrieve and adapt design cases. *Knowledge-Based Systems*, 9(2) : 127–135, 1996.
- [12] B. Smyth et M. T. Keane. Adaptation-Guided Retrieval: Questioning the Similarity Assumption in Reasoning, *Artificial Intelligence*, 102(2) : 249–293, 1998.
- [13] R. B. Sovat et ACPLF de Carvalho. Retrieval and Adaptation of Cases Using an Artificial Neural Network. *Workshop on Soft Computing in Case-Based Reasoning (in ICCBR'01)*, Vancouver, Canada, 2001.
- [14] « IMS Learner Information Packaging Best Practice & Implementation Guide » <http://www.msglobal.org/profiles/lipbest01a.html>, Date de dernière mise à jour : 02/05/2006.

Bibliographie

- [Aha et al., 2001] Aha D.W., Breslow L.A., Muñoz-Avila H, 2001. Conversational case-based reasoning, *Applied Intelligence*, vol 14, pp. 9-32.
- [Alliot et al., 2002] J.-M. Alliot, T.Schiex, P. Brisset, F. Garcia. *Intelligence artificielle & informatique théorique*. Editions Cépadues. 2ième édition 2002. ISBN : 2.85428.578.6.
- [Anand et al., 1998] S. S. Anand, D. Patterson, J. G Hugues. et D. Bell (1998). *Discovering Case Knowledge Using Data Mining*. In X. WU, K. Ramamohanarao & K. B. KORB, Eds., Research and Development in Knowledge Discovery and Data Mining, Second Pacific-Asia Conference, PAKDD-98, Melbourne, Australia, April 15-17, 1998, Proceedings, Lecture Notes in Artificial Intelligence 1394, p. 25–35. Springer.
- [D'Aquin et al. 04] M. d'Aquin, S. Brachais, J. Lieber et A. Napoli (2004). *Vers une acquisition automatique de connaissances d'adaptation par examen de la base de cas- une approche fondée sur des techniques d'extraction de connaissances dans des bases de données*. 12ème Atelier de Raisonnement à Partir de Cas – RÀPC'04, Université Paris Nord, Villetaneuse, France, p. 41-52.
- [Bellman 1978] R. Bellman. *An Introduction to Artificial Intelligence: Can Computers Think?* Boyd & Fraser Publishing Company, San Francisco, 1978.
- [Bergmann et al., 1995] R. Bergmann et W. Wilke. *Building and refining abstract planning cases by change of representation language*. Journal of Artificial Intelligence Research, 3:53–118, 1995.
- [Bergmann et al., 1996] R. Bergmann et W. Wilke. *PARIS:Flexible PlanAdaptation by Abstraction and Refinement*. Dans *Workshop on Adaptation in Case-Based Reasoning*, ECAI-96, rédacteurs A.VoS, R.Bergmann et B.Bartsch-Sporl, Budapest, Hungary, August 1996.
- [Bonnet et al., 1986] : Alain Bonnet, Jean-Michel Truong and Jean-Paul Haton. *Systèmes Experts*. InterEditions, Paris, 1986.
- [Carbonell 1983] J. G. Carbonell (1983). *Learning by analogy: Formulating and generalizing plans from past experience*. In *Machine Learning*, chapter 5, Tioga Press, p. 137–161.
- [Carbonell 1986] J. G. Carbonell (1986). *Derivational analogy; A theory of reconstructive problem solving and expertise acquisition*. In R.S. Michalski, J.G. Carbonell, T.M. Mitchell (eds.): *Machine Learning - An artificial Intelligence Approach*, Vol.II, Morgan Kaufmann, p. 371-392.
- [Cho et al., 1999] K. D. Cho, J. P. Lee, K. T. Kim et S. C. Hwang (1999). *A Learning of Adaptation Knowledge for Case-Based Reasoning System*. In M. H. HAMZA, Ed., Proceedings of the 17th IASTED International Conference Applied Informatics, p. 673–675, Innsbruck, Austria: ACTA Press, Anaheim, Calgary, Zurich.
- [Corchado et al. 2001] J. Corchado et B. Lees (2001). *Adaptation of cases for case-based forecasting with neural network support*. In S. K. Pal, T. S. Dillon et D. S. Yeung, Eds., *Soft Computing in Case-Based Reasoning*, chapter 13, p. 293–319. Springer.

- [Cordier et al., 2006] Cordier A., Fuchs B. & Mille A. (2006). De l'Ingénierie à l'Apprentissage des connaissances d'adaptation en Raisonnement à Partir de Cas. Journées ingénierie des connaissances - IC'2006, p. 41-50, Toulouse.
- [de Silva Garza et al., 1999] A. G. de Silva Garza, M. L. Maher (1999). *An evolutionary approach to case adaptation*. In: Proceedings of Third International Conference on Case-Based Reasoning ICCBR-99 , Monastery Seon, Munich, Germany, p162–172.
- [Fuchs et al., 1995] B. Fuchs, B. Chiron et A. Mille (1995a). Decision Helping in Process Supervision: the Padim Project. In Workshop on Practical Development Strategies for Industrial strength Case Based Reasoning Applications, 16th International Conference on Artificial Intelligence. Montréal, Canada.
- [Gosselin 1996] B. Gosselin. *Application de réseaux de neurones artificiels à la reconnaissance automatique de caractères manuscrits*. Thèse de doctorat à la Faculté Polytechnique de Mons, 1996.
- [Hammond 1989] K. J. Hammond (1989). *CHEF*. In C. Riesbeck et R. C. Schank, Eds., Inside Case-Based Reasoning, chapter 6, p. 165–212. Lawrence Erlbaum.
- [Hanney et al., 1996] K. Hanney et M. T. Keane (1996). *Learning Adaptation Rules From a Case-Base*. In I. SMITH & B. FALTINGS, Eds., Advances in Case-Based Reasoning – Third European Workshop, EWCBR'96, LNAI 1168, p. 179–192: Springer Verlag, Berlin.
- [Hastings et al., 1995] J. D. Hastings, L. K. branting et J. A. Lockwood (1995). *Case adaptation using an incomplete causal model*. In M. VELOSO & A. AAMODT, Eds., Case-Based Reasoning Research and Development – First International Conference, ICCBR'95, Sesimbra, Portugal, Lecture Notes in Artificial Intelligence 1010: Springer Verlag, Berlin, p. 181–192.
- [Haykin 1994] S. Haykin. *Neural Networks - A comprehensive Foundation*. Macmillan College Publishing Company, IEEE PRESS, 1994.
- [Hérault et al., 1994] J. Hérault, C. Jutten (1994). *Réseaux neuronaux et traitement du signal*, Hermes.
- [Jackson 1986]: P. Jakson. *Introduction to expert systems*. Addison-Wesley, 1986.
- [Jarmulak et al., 2001] J. Jarmulak, S. Craw et R. Rowe (2001). *Using Case-Base Data to Learn Adaptation Knowledge for Design*. In Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI'01), p. 1011–1016: Morgan Kaufmann, Inc.
- [Jodouin 1994] J. F. Jodouin. *Les réseaux neuromimétiques*. Hermès, 1994.
- [Koehler 1996] J. Koehler. *Planning from Second Principles*. Artificial Intelligence, 87:145–186, 1996.
- [Kolodner 1982] J. Kolodner. *Knowledge based self organizing memory*. In R. C. Schank. Editor. Dynamic Memory . A Theory of Reminding and Learning in Computer and People. Volume 2. Cambridge university press, Cambridge.
- [Kurzweil 1990] R. Kurzweil. *The Age of Intelligent Machines*. MIT Press, Cambridge, Massachusetts, 1990.

- [Lamontagne 2001] L. Lamontagne. Raisonement à base de cas textuel pour la réponse automatique au courrier électronique. Présentation de sujet de thèse. Université de Montréal. Montréal (Canada), Novembre 2001.
- [Lamontagne et al., 2002] L. Lamontagne, G. Lapalme. Raisonement à partir de cas textuels - état de l'art et perspectives. Montréal (Québec), 2002.
- [Lamontagne 2004] L. Lamontagne. Une approche CBR textuel de réponse au courrier électronique. Thèse de Ph.D. Université de Montréal, 2004.
- [Leake et al., 1997] D. B. Leake, A Kinley. et D. Wilson. (1997). *Case-Based Similarity Assessment: Estimating Adaptability from Experience*. Fourteenth National Conference on Artificial Intelligence, AAAI Press, Menlo Park, CA, p. 674-679.
- [Lee 2003] M. LEE (2003). *A study of an automatic learning model of adaptation knowledge for case base reasoning*. Inf. Sci., 155(1-2), 61-78.
- [Lefébure et al., 2001] R. Lefébure, G. Venturi. *Data Mining*. deuxième édition, Eyrolles, 2001.
- [Leroy 1996] B. Leroy. *Reconnaissance de visage*. Thèse de doctorat. paris-six dauphine. 1996.
- [Lieber et al., 1996] Lieber J et Napoli A 1996. *Adaptation of Synthesis Plans in Organic Chemistry*. In Workshop on Adaptation in Case Based Reasoning. ECAI-96, Budapest, Hungary.
- [Lieber 1997] J. Lieber, 1997. *Raisonement à partir de cas et classification hiérarchique. Application à la planification de synthèse en chimie organique*. Thèse d'université. Université Henri Poincaré Nancy 1.
- [Lieber et al. 2001] Lieber J., Bey P., Boisson F., Bresson B., Falzon P., Lesur A., Napoli A., Rios M. & Sauvagnac C. (2001). Acquisition et modélisation de connaissances d'adaptation, une étude pour le traitement du cancer du sein, Journées ingénierie des connaissances. Journées ingénierie des connaissances - IC'2001, Presses Universitaires de Grenoble, Grenoble, France, p. 409-426.
- [Lieber et al., 2003] J. Lieber, M. d'Aquin, P. Bey, A. Napoli, M. RIOS et C. Sauvagnac (2003). *Adaptation knowledge acquisition and modeling, a study for breast cancer treatment*. In Artificial Intelligence in Medecine Europe (AIME'03), October 2003: Springer.
- [Lieber et al., 2004] Lieber J., d'Aquin M., Brachais S. & Napoli A. (2004). Une étude comparative de quelques travaux sur l'acquisition de connaissances d'adaptation pour le raisonnement à partir de cas. In Sylvie Salotti, Rushed Kanawati Et Farida Zehraoui, Ed., actes 12ème atelier raisonnement à partir de cas, RàPC'04.
- [Luo et al., 1989] R. C. Luo et M. G. Kay. *Multisensor integration and fusion in intelligent systems*. Man, and Cybernetics, pages : 901-931, année 1989.
- [Milgram 1993] M. Milgram. *Reconnaissance Des Formes. Méthodes numériques et connexionnistes*. Edition Armand Colin , 1993.
- [Mille 1995a] Mille A (1995a). Modele conceptuel du raisonnement basé sur les cas. In Bichindaritz, I, editor. 4ième Séminaire Français sur le Raisonement à Partir de Cas. Université René descartes, Paris V.

- [Mille 1995b] A. Mille (1995b). Raisonement basé sur l'expérience pour coopérer à la prise de décision, un nouveau paradigme en supervision industrielle. Thèse d'université. Université Jean Monnet, Saint-Étienne.
- [Newell et al., 1981] A. Newell et H. A. Simon. *Physical symbol systems*. In D. A. Norman, editor, *Perspectives on cognitive science*, pages 3785. Norwood, NJ Ablex Publishing Corp., 1981. ISBN: 0-89391-071-6, ISBN: 0-89859-106-6.
- [Purvis et al., 1998] L. Purvis, P. Pu. *Composer: A case-based reasoning system for engineering design*. In: *Robotica*. (1998) 285–295
- [Rakoto 2004] H. Rakoto. *Intégration du Retour d'Expérience dans les processus industriels*. Application à Alstom Transport. Thèse de doctorat à l'Institut National Polytechnique de Toulouse.
- [Rapaport 1983] W J. Rapaport. Cognitive science. In A. Ralston and E. D Reilly, Editors, *Encyclopedia of computer science and engineering*. Van Nostrand Reinhold, 1983. ISBN: 0-442-24496-7.
- [Rich et al., 1991] E. Rich et K. Knight. *Artificial Intelligence*. McGraw-Hill, New-York, second edition, 1991.
- [Royer 1990] Jean-Claude ROYER. *MANDRIN: Un système d'apprentissage pour l'aide au réglage d'un instrument*. Thèse de Doctorat : Institut National Polytechnique de Grenoble, 1990.
- [Rumelhart et al., 1986] D. E. Rumelhart, G. E. Hinton, et R. J Williams. *Learning internal representations by error propagation*. In D. E. Rumelhart and J. McClelland, editors, *Parallel Distributed Processing*. MIT Press, Cambridge, MA, 1986.
- [Salton et al., 1984] Salton G., McGill M., 1984. *Introduction to Modern Information Retrieval*, McGraw-Hill Book Company.
- [Schank 1982] R. Schank (1982). *Dynamic memory; a theory of reminding and learning in computers and people*. Cambridge University Press.
- [Smyth et al., 1995] B. Smyth et M. T. Keane (1995). *Remembering to forget: A competence-preserving deletion policy for CBR systems*. In 14th Joint Conference on Artificial Intelligence, IJCAI-95, p. 377–382, Montréal, Canada.
- [Smyth et al., 1998] B. Smyth et M. T. Keane (1998). *Adaptation-Guided Retrieval: Questioning the Similarity Assumption in Reasoning*, *Artificial Intelligence*, 102(2): 249.293.
- [Wikipédia] http://fr.wikipedia.org/wiki/Réseau_de_neurones
- [Wilke et al., 1997] W. Wilke, I. Vollrath, K. D. Althoff et R. Bergmann (1997). *A Framework for Learning Adaptation Knowledge Based on Knowledge Light Approaches*. In *Proceedings of the Fifth German Workshop on Case-Based Reasoning*, p. 235–242.
- [Wooldrige et al., 1995]

Résumé

Le raisonnement à partir de cas (RàPC) est un paradigme issu de l'intelligence artificielle qui permet la mise en place d'un système à base de connaissances. Il consiste à résoudre un nouveau problème en se remémorant et en adaptant un ou plusieurs cas passés déjà résolus. L'adaptation constitue l'étape la plus importante du processus de RàPC. Elle nécessite l'acquisition d'un ensemble de connaissances spécifiques généralement très difficiles à modéliser. Dans cette thèse, nous proposons une technique d'adaptation des cas qui est basée sur l'utilisation de réseau de neurones. On fournit à l'entrée du réseau le nouveau problème rencontré et on récupère à sa sortie la solution adaptée. Cette technique présente deux principaux avantages: (1) l'apprentissage des réseaux de neurones se fait avant le processus de raisonnement ; (2) les réseaux générés peuvent être réutilisés dans les futures adaptations. Elle sera appliquée à l'adaptation des scénarios d'apprentissage pédagogique, décrits sous forme de suites d'activités, aux apprenants décrits avec le modèle IMS-LIP.

Mots clés: Raisonnement à partir de cas (RàPC), adaptation, acquisition de connaissances d'adaptation (ACA), réseaux de neurones, scénario d'apprentissage, apprenant.

Abstract

Case-based reasoning (CBR) is a paradigm of Artificial Intelligence that allows setting-up of a knowledge-based system. It tries to resolve a new problem by retrieving and adapting past cases which are already resolved. The adaptation constitutes the most important step of the CBR process. It requires the acquirement of a set of specific knowledge which is very difficult to model. In this thesis, we present a case adaptation technique which is based on neural networks. We present at the input of the network the new problem and we recover at the output the adapted solution. This technique presents two main advantages: (1) the learning of the neural network is realized before the CBR process; (2) the generated networks can be reused in the future adaptations. This technique will be applied to the learning scenarios adaptation, described with sequence of activities, to the learners described with IMS-LIP.

Keywords: Case based reasoning (CBR), adaptation, adaptation knowledge acquisition (AKA), neural networks, learning scenario, learner.

ملخص:

الاستدلال عن طريق الحالة هو طريقة من طرق الذكاء الاصطناعي و الذي يسمح بتجسيد نظام مبني على المعلومات. إنه يركز في حل مشكلة جديدة على استعمال و تكييف حلول المشاكل القديمة. التكيف يشكل المرحلة الأكثر أهمية في سياق الاستدلال عن طريق الحالة. هذه المرحلة تستلزم اقتناء مجموعة من المعلومات الخاصة والتي هي صعبة التجسيد.

خلال هذه الأطروحة, نستعرض طريقة جديدة لتكييف الحالات و التي تعتمد على استعمال شبكة العصبونات الاصطناعية. نعرض عند مدخل الشبكة المشكل الحالي و نسترجع عند المخرج الحل المكيف. هذه الطريقة لها فائدتان أساسيتان: (1) تمرن شبكات العصبونات يكون قبل مرحلة الاستدلال, (2) شبكات العصبونات الاصطناعية يمكن استعمالها خلال التكييفات المستقبلية. هذه الطريقة سوف نستعملها لتكييف نماذج التعليم التربوية الممتلة بسلاسل من النشاطات, لمتعلمين ممثلين بالنموذج إمس-ليب. **الكلمات المفتاح:** الاستدلال عن طريق الحالة, تكييف, اقتناء معلومات التكيف, شبكة العصبونات, نماذج التعليم, متعلم.