

**HYBRID MODELS OF FUZZY ARTMAP AND Q-  
LEARNING FOR PATTERN CLASSIFICATION**

**FARHAD POURPANAH NAVAN**

**UNIVERSITI SAINS MALAYSIA**

**2015**

**HYBRID MODELS OF FUZZY ARTMAP AND Q-LEARNING FOR  
PATTERN CLASSIFICATION**

**By**

**FARHAD POURPANAH NAVAN**

**Thesis submitted in fulfilment of the requirements  
for the degree of  
Doctor of Philosophy**

**October 2015**

## ACKNOWLEDGMENTS

All praises are due to the Creator without Whom I wouldn't have finished this PhD research work.

I would like to express my gratitude to my first main supervisor and current co-supervisor, Professor Lim Chee Peng, whatever I cannot explain with words, for his great advice, patience, warm concern, valuable suggestions, and endless support that has helped me to complete this dissertation. In addition, I would like to thank him for his technical and editorial advices that have been essential to complete this research and has taught me countless lessons in general. It has been an honour to work with him.

I would like to thank to my second main-supervisor, Assoc. Prof. Dr. Umi Kalthum Ngah, for her advice and encouragement and sharing her experiences. Also, I would like to express the deepest appreciation to my current main-supervisor, Assoc. Prof. Dr. Junita Mohamad Saleh, for her inspiration and kindness. I appreciate the time she has spent on reviewing this thesis for final touch-ups.

I would like to acknowledge the School of Electrical and Electronic Engineering, USM for providing me the necessary equipment as well as the Graduate Assistant (GA) support, which has made this research possible. I also would like to thank all staff at the School for their support.

My final thanks go to my family members who have always stood beside me, in particular, my parents, Mr. Heidar Pourpanah Navan and Ms. Raheleh Pourpanah Navan, for their endless love, support and encouragement throughout my life. In addition, I would like to thank my brothers, Mr. Behrad and Mr. Taha, as well as my sister Miss Sepideh. Without their prayers, support, and encouragement, I would have never been able to complete my work. I owe them more than I could ever repay.

## TABLE OF CONTENTS

Acknowledgment.....	ii
Table of Contents.....	iii
List of Tables.....	vii
List of Figures.....	x
List of Abbreviations.....	xii
List of Symbols.....	xv
Abstrak.....	xvii
Abstract.....	xviii

### CHAPTER 1 - INTRODUCTION

1.1 Background of the study.....	1
1.2 Computational Intelligence.....	3
1.3 Reinforcement Learning (RL).....	4
1.4 Problems and Motivations.....	5
1.5 Research Objectives.....	9
1.6 Thesis Outline.....	10

### CHAPTER 2 - LITERATURE REVIEW

2.1 Introduction.....	12
2.2 Artificial Neural Network.....	12
2.2.1 Fuzzy ARTMAP.....	17
2.3 Reinforcement Learning.....	21
2.3.1 Control and Robotic Tasks.....	22

2.3.2	Computer Games and Industrial Tasks.....	25
2.3.3	Clustering and Classification tasks.....	26
2.4	Feature Selection and Rule Extraction.....	27
2.5	Ensemble Methods.....	31
2.6	Multi-Agent Systems.....	37
2.7	Summary.....	41

## CHAPTER 3 - DEVELOPMENT OF FUZZY ARTMAP AND REINFORCEMENT LEARNING HYBRID MODEL

3.1	Introduction.....	42
3.2	Research Methodology.....	43
3.3	Dynamics of Fuzzy ARTMAP and Reinforcement Learning.....	46
3.3.1	Dynamics of Fuzzy ARTMAP.....	46
3.3.2	Reinforcement Learning.....	54
3.4	Proposed QFAM Model.....	56
3.4.1	Estimation of Q-value.....	57
3.4.2	Selection of the Winning Node.....	58
3.4.3	Proposed QFAM Algorithm.....	59
3.4.4	A Numerical Example.....	61
3.5	Experimental Studies.....	67
3.5.1	The Circle-in-the-Square Problem.....	68
3.5.2	Experiments with the UCI Data Sets.....	72
3.6	Summary.....	75

## CHAPTER 4 - IMPROVED FUZZY ARTMAP AND REINFORCEMENT

### LEARNING MODELS

4.1	Introduction.....	77
4.2	Genetic Algorithm.....	78
4.3	Proposed Two-Stage Classifier with Rule Extraction.....	79
	4.3.1 Stage 1: QFAM with Pruning.....	80
	4.3.2 Stage 2: QFAM with Rule Extraction.....	80
4.4	Experimental Studies.....	88
	4.4.1 Experiments with UCI Data Sets.....	89
	4.4.2 Experiments with the KEEL data sets.....	93
	4.4.3 Rule Extraction Analysis.....	99
4.5	Summary.....	103

## CHAPTER 5 - DEVELOPMENT OF A MULTI-AGENT CLASSIFIER SYSTEM

5.1	Introduction.....	105
5.2	Agents and Multi-Agent Systems.....	106
5.3	Trust-Negotiation-Communication (TNC).....	107
5.4	The proposed QMACS Model.....	108
	5.4.1 Estimating the Initial Trust Measurement of Agents.....	110
	5.4.2 Updating the Trust value for each test sample.....	113
	5.4.3 Summary of the Proposed QMACS Model.....	114
5.5	Experimental studies.....	116
	5.5.1 Case Study I.....	117
	5.5.2 Case Study II.....	121
5.6	Summary.....	125

## CHAPTER 6 - REAL-WORLD APPLICATIONS

6.1	Introduction.....	127
6.2	Sensitivity and Specificity.....	127
6.3	Feature Extraction.....	128
6.4	Real-World Case Studies.....	129
	6.4.1 Case Study 1.....	130
	6.4.2 Case Study 2.....	137
6.5	Summary.....	142

## CHAPTER 7 - CONCLUSION AND FUTURE WORK

7.1	Conclusion.....	144
7.2	Contributions of the research.....	146
7.3	Suggestions for Future Work.....	147
	7.3.1 Using Different CI-Based Networks.....	148
	7.3.2 Using QFAM-based models as Function Approximation.....	148
	7.3.3 Expanding QFAM-based models.....	148
	7.3.4 Enhancing the QMACS model.....	149

REFERENCES.....	150
-----------------	-----

LIST OF PUBLICATIONS.....	172
---------------------------	-----

## LIST OF TABLES

	<b>Page</b>
Table 2.1 Summary of the FAM variants	20
Table 2.2 Summary of RL-based applications	27
Table 2.3 Summary of reviewed rule extraction and feature selection methods	30
Table 2.4 Summary of reviewed neural network ensemble models	35
Table 2.5 Summary reviewed MAS models	40
Table 3.1 Details of UCI data sets	67
Table 3.2 Test accuracy rates for the circle-in-the-square problem	69
Table 3.3 Test accuracy rates for the noisy circle-in-the-square problem	71
Table 3.4 Test accuracy rates (%) for the UCI data sets	73
Table 3.5 Classification accuracy results for the UCI data sets	75
Table 4.1 Terms related to the GA	78
Table 4.2 The original prototype node all with its possible “open” prototypes (dc indicates “don’t care”)	82
Table 4.3 Details of UCI data sets	89
Table 4.4 Test accuracy rate (%) for the UCI data sets with 95% confidence intervals	91
Table 4.5 Mean accuracy (%) for the UCI data sets	92
Table 4.6 Table 4.6: Mean accuracy (%) with standard deviation (SD) for the UCI data sets	93
Table 4.7 Details of the KEEL data sets	93
Table 4.8 The accuracy rates (%) for the Iris data set	94
Table 4.9 The accuracy rates (%) for the Heart (Statlog) data set	94
Table 4.10 The accuracy rates (%) for the Wine data set	95
Table 4.11 The accuracy rates (%) for the Pima Indian Diabetes (PID) data	95



	set	
Table 4.12	The accuracy rates (%) for the Iris data set with different noise levels	98
Table 4.13	The accuracy rates (%) of the Heart data set with different noise levels	98
Table 4.14	The accuracy rates (%) of the PID data set with different noise levels	98
Table 4.15	The accuracy rates (%) for the Wine data set with different noise levels	98
Table 4.16	Details of the (a) input features and (b) target classes for the PID data set	100
Table 4.17	The extracted rule with the “don’t care” (dc) antecedent for the PID data set	100
Table 4.18	Details of the (a) input features and (b) target classes for the Iris data set	101
Table 4.19	The extracted rule with the “don’t care” (dc) antecedent for the Iris data set	101
Table 4.20	Details of the (a) input features and (b) target classes for the Glass data set	102
Table 4.21	The extracted rule with the “don’t care” (dc) antecedent for the Glass data set	102
Table 5.1	Details of UCI data sets	116
Table 5.2	The accuracy rates (%) for the four-circle-in-the-square problem	120
Table 5.3	The accuracy rates (%) for the four-circle-in-the-square	121
Table 5.4	The accuracy rates (%) for the UCI data sets	123
Table 5.5	The accuracy rates (%) for the waveform problem	124
Table 5.6	The accuracy rates for the LED problem	124
Table 5.7	The mean accuracy rates (%) with standard deviations for the Waveform problem	125
Table 5.8	The error rates for the LED problem	125
Table 6.1	Time domain features	129

Table 6.2	Details of human motion data set	130
Table 6.3	The accuracy rates (%) of individual models with 95% confidence intervals for human motion detection	131
Table 6.4	The number of prototypes before and after pruning with 95% confidence intervals	132
Table 6.5	The accuracy rates of QMACS and its agent teams with 95% confidence intervals for human motion detection	133
Table 6.6	The overall accuracy rates (%) of QFAM, Pruned QFAM and QFAM-GA with 95% confidence intervals for motor fault detection	138
Table 6.7	The number of prototypes before and after pruning with 95% confidence intervals for motor fault detection	138
Table 6.8	The accuracy rates (%) of QMACS and its agent teams with 95% confidence intervals for motor fault detection	139

## LIST OF FIGURES

	<b>Page</b>
Figure 2.1 Different types of FAM-based models	17
Figure 3.1 Research methodology	46
Figure 3.2 The structure of FAM (adapted from Carpenter et al. (1992))	47
Figure 3.3 The block diagram of proposed stages of QFAM algorithm	59
Figure 3.4 Test accuracy rates of QFAM (with 95% confidence factors indicated as error bars) and FAM	70
Figure 3.5 The decision boundaries created by QFAM with respect to different training data sizes	70
Figure 3.6 The number of prototype nodes created by QFAM for the noisy circle-in-the-square problem	72
Figure 4.1 Block diagram of the proposed two-stage QFAM-GA model	80
Figure 4.2 Flowchart of the GA rule selection process	85
Figure 4.3 Flowchart of the proposed QFAM-GA model	87
Figure 4.4 The number of prototype nodes before and after pruning	91
Figure 4.5 The number of prototypes for the Iris data set before and after pruning	96
Figure 4.6 The number of prototypes for the Heart data set before and after pruning	96
Figure 4.7 The number of prototypes for the Wine data set before and after pruning	97
Figure 4.8 The number of prototypes for the PID data set before and after pruning	97
Figure 5.1 The TNC reasoning scheme (adapted from Haider et al. 2006)	107
Figure 5.2 The proposed QMACS model	109
Figure 5.3 A block diagram of the proposed QMACS operations	110
Figure 5.4 The circle-in-the-square problem with 10000samples	118

Figure 5.5	Test accuracy rates of QMACS and its team agents for the circle-in-the-square problem	119
Figure 5.6	The four-circle-in-the-square problem with 2000 samples for (a) noise-free and (b) noisy data set	120
Figure 5.7	The accuracy rates (%) of QMACS and its team agents for various UCI data sets	123
Figure 6.1	The accuracy rates of QMACS and its agent teams with 95% confidence intervals for human motion detection in the presence of noise	134
Figure 6.2	The accuracy rates of QMACS and its agent teams with 95% confidence intervals for human motion detection in the presence of noise	135
Figure 6.3	The SENS results of QMACS and its team agents for human motion detection	136
Figure 6.4	The SPEC results of QMACS and its team agents for human motion detection	136
Figure 6.5	The accuracy rates of QMACS and its agent teams with 95% confidence intervals for motor fault detection	140
Figure 6.6	The number of prototypes before and after pruning with 95% confidence intervals in the presence of noise	141
Figure 6.7	Figure 6.7: The SENS results of QMACS and its team agents for the motor fault detection data set with 95% confidence intervals	142
Figure 6.8	The SPEC results of QMACS and its team agents for the motor fault detection data set with 95% confidence intervals	142

## LIST OF ABBREVIATIONS

ADaBoost	Adaptive Boosting
ACC	Accuracy
ACLA	Actor-Critic Learning Automata
ACS	Adaptive Classification System
ART	Adaptive Resonance Theory
ARTMAP	Adaptive Resonance Theory Mapping
ANN	Artificial Neural Network
BAR	Bayesian ARTMAP
BNN	Backpropagation Neural Network
CNeT	Competitive Neural Tress
CAFE	Collaborative Agents for Filtering E-mails
CVNN	Complex-Valued Neural Network
CI	Computational intelligence
CAMAS	Context-Aware Multi-Agent System
CF	Crest factor
CFS	Correlation-based Feature Selection
DC	Dendritic Cell
DARL	Direct Reinforcement Adaptive Learning
DDA	Dynamic Decay Adjustment
DPSO	Dynamic Particle Swarm Optimization
DP	Dynamic Programming
EFMM	Enhanced Fuzzy Min-Max
EA	Evolutionary Algorithm

ESPEC	Specificity
FAM	Fuzzy ARTMAP
FI	Fuzzy Integral
FMM	Fuzzy Min-Max
FS	Fuzzy System
GA	Genetic Algorithm
GARE	Generalized Analytic Rule Extraction
GPI	Generalized Policy Iteration
GPM	Global Power Manager
GRG	Greedy Rule Generation
HIF	High Impedance Faults
HHONC	Hybrid Higher Order Neural Classifier
IF	Impulse factor
IRL	Inverse Reinforcement Learning
KGS	Kanas Geological Survey
KLSPi	Kernel-based Least Square Policy Iteration
KNN	K-Nearest Neighbour
KEEL	Knowledge Extraction based on Evolutionary Learning
LF	Latitude factor
LSPI	Least Square Policy Iteration
LP	Linear Programming
LPM	Local Power Manager
LTM	Long Term Memory
MDP	Markov Decision Process

MACS	Multi-Agent Classifier System
MATC	Multi-Agent Text Classification
MAS	Multi-Agent System
MLP	Multi-Layered Perceptron
MFS	Multiple Feature Subsets
MC	Monte Carlo
NNE	Neural Network Ensemble
OELM	Online Extreme Learning Machine
PQ	Power Quality
PNN	Probabilistic Neural Network
POPTVR	Pseudo-Outer Product Truth Value Restriction
QFAM	Q-learning Fuzzy ARTMAP
RBS	Radial Basis Function
RNCL	Regularized Negative Correlation Learning
RL	Reinforcement Learning
SCM	Separability-Correlation Measure
SENS	Sensitivity
SF	Shape factor
SVM	Support Vector Machine
TD	Temporal Difference
TNC	Trust-Negotiation and Communication
VE	Virtual Enterprise
WOL	Web Ontology Language

## LIST OF SYMBOLS

$\alpha$	Choice parameter
$\beta$	Learning parameter
$\rho$	Vigilance parameter
$\xi$	Learning rate
$r(j)$	Reinforcement signal
$\lambda$	Weighting factor
$\gamma$	Discount factor
$vig(j)$	Vigilance test value of node $j$
$Q\_values_j$	Q-value of node $j$
$strength(j)$	Strength of node $j$
$V^\pi(s,a)$	The value function of action $a$ in state $s$ under policy $\pi$
$A$	Complement code of input $a$
$B$	Complement code of input $b$
$\delta$	A small positive value
$\rho_{ab}$	The map-field vigilance parameter
$N_a$	The number of nodes in $ART_a$
$N_b$	The number of nodes in $ART_b$
$\wedge$	Fuzzy AND operator
$f(s)$	Fitness function
$NCP(s)$	Number of correctly classified patterns
$ S $	Number of chosen input features
$N_{pop}$	Number of population size



$P(S)$	Probability of selecting string $S$
$\psi$	Population
$Q$	Quantization level
$R_j$	Rule number $j$
$CM^k$	Confusion matrix of $k$ -th agent
$n_{ij}^k$	Total number of input samples of class $i$ predicted as class $j$ by $k$ -th agent
$e_k$	$k$ -th agent
$bel(i)$	Belief function of $i$ -th agent
$initial\_trust(i)$	Initial trust of $i$ -th agent
$trust(i)$	Trust of $i$ -th agent
$Q\_Value_{C_m}^k$	Mean of all Q-values that are related to class $m$ of agent $k$
$Q_{C_m z}$	Q-value of the $z$ -th prototype of class $m$

# **MODEL HIBRID PEMETAAN TEORI RESONAN ADAPTIF KABUR DAN PEMBELAJARAN-Q UNTUK PENGELASAN CORAK**

## **ABSTRAK**

Pengelasan corak adalah salah satu isu utama dalam pelbagai tugas pencarian data. Dalam kajian ini, fokus penyelidikan tertumpu kepada reka bentuk dan pembinaan model hibrid yang menggabungkan rangkaian neural Teori Resonan Adaptif (ART) terselia dan model Pembelajaran Pengukuhan (RL) untuk pengelasan corak. Secara khususnya, rangkaian ARTMAP Kabur (FAM) dan Pembelajaran-Q dijadikan sebagai tulang belakang dalam merekabentuk dan membina model-model hibrid. Satu model QFAM baharu terlebih dahulu diperkenalkan bagi menambahbaik prestasi pengelasan rangkaian FAM. Strategi *pruning* dimasukkan bagi mengurangkan kekompleksan QFAM. Bagi mengatasi isu ketidak-telusan, Algoritma Genetik (GA) digunakan bagi mengekstrak hukum kabur *if-then* daripada QFAM. Model yang terhasil iaitu QFAM-GA, dapat memberi ramalan berserta dengan huraian dengan hanya menggunakan bilangan antisiden yang sedikit. Bagi menambahkan lagi kebolehtahanan model-model Q-FAM, penggunaan sistem agen-pelbagai telah dicadangkan. Hasilnya, model gugusan QFAM berasaskan agen dengan ukuran percaya dan kaedah rundingan baharu telah dicadangkan. Pelbagai jenis masalah tanda-aras telah digunakan bagi penilaian model-model gugusan dan individu berasaskan QFAM. Hasilnya telah dianalisa dan dibandingkan dengan FAM serta model-model yang dilaporkan dalam kajian terdahulu. Sebagai tambahan, dua daripada masalah dunia-nyata digunakan bagi menunjukkan kebolehan praktikal model hibrid. Keputusan akhir menunjukkan keberkesanan modul berasaskan QFAM dalam menerajui tugas-tugas pengelasan corak.

# **HYBRID MODELS OF FUZZY ARTMAP AND Q-LEARNING FOR PATTERN CLASSIFICATION**

## **ABSTRACT**

Pattern classification is one of the primary issues in various data mining tasks. In this study, the main research focus is on the design and development of hybrid models, combining the supervised Adaptive Resonance Theory (ART) neural network and Reinforcement Learning (RL) models for pattern classification. Specifically, the Fuzzy ARTMAP (FAM) network and Q-learning are adopted as the backbone for designing and developing the hybrid models. A new QFAM model is first introduced to improve the classification performance of FAM network. A pruning strategy is incorporated to reduce the complexity of QFAM. To overcome the opaqueness issue, a Genetic Algorithm (GA) is used to extract fuzzy *if-then* rules from QFAM. The resulting model, i.e. QFAM-GA, is able to provide predictions with explanations using only a few antecedents. To further improve the robustness of QFAM-based models, the notion of multi agent systems is employed. As a result, an agent-based QFAM ensemble model with a new trust measurement and negotiation method is proposed. A variety of benchmark problems are used for evaluation of individual and ensemble QFAM-based models. The results are analyzed and compared with those from FAM as well as other models reported in the literature. In addition, two real-world problems are used to demonstrate the practicality of the hybrid models. The outcomes indicate the effectiveness of QFAM-based models in tackling pattern classification tasks.

# CHAPTER 1

## INTRODUCTION

### 1.1 Background of the study

It is generally recognized that pattern recognition is a basic function of human cognition (Wang, 2008). Since the last few decades, human's brain has attracted great attention in both experimental and theoretical aspects. The results have demonstrated that the brain has a tremendous parallel architecture that contains many individual neurons with synapses (interconnections). Human's brain can easily understand a particular situation, recognize face or speech, and also is able to receive patterns from sensing organs and convert them into helpful information to make decisions (Cenggoro et al., 2014). Indeed, humans encounter plenty of recognition tasks daily and make the corresponding decisions unconsciously. By exploiting the technology of digital computers and developing the necessary machine learning and artificial intelligence algorithms, it is now possible to utilize computers to mimic the performance of human's brain. As a result, many investigations have been conducted to tackle pattern recognition problems.

To solve pattern recognition problems by using a computerized system, it is essential to have appropriate algorithms that are able to exploit proper features from received information or data to recognize patterns. In general, there are four main stages in developing a pattern recognition system. They are (Rosenfeld & Wechsler, 2000): (i) *Data Acquisition and Collection*, (ii) *Feature Extraction and Representation*, (iii) *Similarity Detection and Pattern Classification*, and (iv) *Performance Evaluation*.

Over the years, many methodologies have been proposed for pattern classification. Statistical methods are one of the earliest methodologies for pattern classification. These include the discriminatory analysis proposed by Fisher (1936) and Rao (1948). Bayesian decision theorem is another statistical method that has been extensively used to tackle pattern classification problems (Devijver & Kittler, 1982; Duda & Hart, 1973). Nevertheless, statistical methods are inefficient in handling contextual or structural information of patterns, as explained by Pal and Pal (2002). Syntactic techniques, which are related to the theory of formal languages, have been suggested to overcome this problem (Hopcroft & Ullman, 1979). Nevertheless, syntactic technique does not perform well in the presence of noisy data (Pal & Pal, 2002).

Computational intelligence (CI) (Bezdek, 1992; Marks, 1993) is another useful methodology that has been widely applied to solving a variety of applications, e.g. biomedical (Shi & Eberhart, 1998; Yang et al., 2007), mobile robotics (Wang, 2002), healthcare (Tejima, 2003), Web (Zhang, 2005), games (Lucas, 2009), business (Haider & Nishat, 2009), power system (Venayagamoorthy, 2009), control (Wilamowski, 2010), and wireless (Iram et al., 2011). CI has also been extensively employed to tackle pattern classification problems. Generally, CI contains evolutionary algorithms (EAs), Fuzzy Systems (FSs), artificial neural networks (ANNs), and synthesis of these three models with each other and/or other conventional methods (Shi & Eberhart, 1998; Rutkowski, 2008). The focus of this research is on ANNs and other complementary learning methodologies, which include reinforcement learning (RL) and multi-agent system (MAS), for designing and developing efficient and effective pattern classification systems.

In the following sections, a definition of and an introduction to CI are presented. Then, an introduction to RL is provided. The motivations for developing hybrid ANNs and combining them into a MAS are described. The research scope and objectives are presented. Finally, an overview of the thesis organization is given at the end of the chapter.

## 1.2 Computational Intelligence

In this information era, besides the dramatic growth of computer technologies, researchers have designed and developed various intelligent systems that are able to mimic human's behaviours. Analysing the collected data samples and translating them into useful information and subsequently making appropriate decisions is one of the major challenges. To cope with such problems, CI-based models have been devised to operate as useful systems with "humanlike" problem-solving capabilities (Rutkowski, 2008). A definition of CI is provided by Bezdek (1994), is as follows:

*"A system is computationally intelligent when it deals only with numerical (low-level) data, has pattern recognition component, and does not use knowledge in artificial intelligence (AI) sense."*

Another definition of CI is described by Fogel (1995), is as follows:

*"These technologies of neural, fuzzy, and evolutionary systems were brought together under the rubric of computational intelligence, a relatively new trend offered to generally describe*

*methods of computation that can be used to adapt solutions to new problems and do not rely on explicit human knowledge.”*

FSs, EAs, and ANNs are a number of paradigms under the umbrella of CI (Rutkowski, 2008). While CI-based systems have been successfully used to solve problems in different domains, which include medicine (Schizas, 1997), power systems (Pahwa et al., 2003), biological systems (Wu et al., 2007), web design (Liu, Khudkhudia, & Ming, 2008), games (Lucas, 2009), business (Wu, 2010), computer security (Perez et al., 2010), education (Venayagamoorthy, 2010), as well as industrial systems (Sariyildiz et al., 2013), each CI paradigm has its advantages and limitations. As such, hybrid CI models, which consist of two or more CI paradigms, have been introduced to harness the merits of the constituents.

An ANN can be viewed as a mathematical model that processes information based on the principle of a biological neural network (Cenggoro et al., 2014). Since the inception of the first mathematical model of an artificial neuron by McCulloch and Pitts (1947, 1943), many different ANN architectures have been proposed, e.g. Multi-Layered Perceptron (MLP) (Rumelhart et al., 1986), Radial Basis Function (RBF) (Moody & Darken, 1989). A detailed review on ANNs is given in Chapter 2.

### **1.3 Reinforcement Learning (RL)**

RL (Barto & Sutton, 1998) is a methodology that learns from experience by interacting with the environment. It is a semi-supervised learning method that has advantages over supervised learning methods under certain conditions. Unlike supervised learning whereby the target output for each input sample is clearly

known, only minimal information that indicates the appropriateness of the response pertaining to an input sample is available in RL. As such, it does not require detailed knowledge of the target output. There are two main advantages of RL. Firstly, it has the capability of learning on-line in a search-control-learn mode based on previous experiences (Lee et al., 1998). Secondly, it is an effective method when there is little knowledge about what and how to perform a task (Gullapalli, 1990).

RL has been extensively applied as an effective feedback mechanism to tackle control and decision making problems. Among various successful RL applications include cart-pole balancing (Barto et al., 1983), Backgammon game (Tesauro, 1994), and elevator dispatching problem (Barto & Crites, 1996). RL has also been used to improve the performance of many classifiers (Likas & Blekas, 1996; Likas, 2001; Quah et al., 2005). However, RL is not free from limitations, e.g. the exponential growth of its state-space owing to the curse of dimensionality (Lin & Lee, 1994). Such problems have attracted many researchers to work on RL.

#### **1.4 Problems and Motivations**

Many methods have been proposed to solve pattern classification problems, e.g. k-nearest neighbour (Cover & Hart, 1967), naive Bayes classifier (Domingos & Pazzani, 1997), decision tree (Friedl & Brodley, 1997), Support Vector Machine (SVM) (Vapnik, 1995), Artificial Neural Networks (ANNs) (Venkatesan & Balamurugan, 2001). Among them, ANNs have been used as a useful learning model for solving pattern classification tasks (Zhang, 2000), with the capability of handling non-linear as well as noisy data collected from real-world environments.