NEAR EAST UNIVERSITY
LIBRARY
1988 - LEFKOŞA

NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

ARTIFICIAL INTELLIGENCE SYSTEM

Graduation Project

COM – 400

Student: Nader Ibraik

Supervisor: Assoc. Prof. Dr Adnan Khashman

Nicosia - 2003

ACKNOWLEDGEMENTS

First of all I am thanking full to the most gracious "ALLAH" the almighty, who enable me to complete this project

Secondly, I would like to award my supervisor Assoc. Prof. Dr Adnan Khashman for being so operative averring supervise me in this work, and for his overwhelming and limitless help he had done to me.

Thirdly, I will never over look the encourage I had resaved from all my family, specially my parents, my "father" the best father in this world, and the best sweetest woman in the word "mother", for there supporting me and caring for me, I am thankful for them and my family members from brother and sisters.

Fourthly this will be the most important moment in my life that I will settlement my project to the best brother" ZAKY-IBRAIK" because of his encourage me and supporting me in the long of my 4th year study ,I am so thankful for him, my brother I am own you too match.

Finally I like to thank the best friends I had met them in Cyprus Mohammed Asfour and Mohammad Shuqair and Mohammad Qunj who I can't express my feeling in such word that for there suggestion and evolution through out completion my project.

I will never forget them

NEAR EAST UNIVERSITY

NEAR EAST UNIVERSITY
LIBRARY
1988 - LEFKOŞA

Faculty of Engineering

Department of Computer Engineering

ARTIFICIAL INTELLIGENCE SYSTEM

Graduation Project

COM – 400

Student: Nader Ibraik

Supervisor: Assoc. Prof. Dr Adnan Khashman

Nicosia - 2003

ACKNOWLEDGEMENTS

First of all I am thanking full to the most gracious "ALLAH" the almighty, who enable me to complete this project

Secondly, I would like to award my supervisor Assoc. Prof. Dr Adnan Khashman for being so operative averring supervise me in this work, and for his overwhelming and limitless help he had done to me.

Thirdly, I will never over look the encourage I had resaved from all my family, specially my parents, my "father" the best father in this world, and the best sweetest woman in the word "mother", for there supporting me and caring for me, I am thankful for them and my family members from brother and sisters.

Fourthly this will be the most important moment in my life that I will settlement my project to the best brother" ZAKY-IBRAIK" because of his encourage me and supporting me in the long of my 4^{th} year study ,I am so thankful for him, my brother I am own you too match.

Finally I like to thank the best friends I had met them in Cyprus Mohammed Asfour and Mohammad Shuqair and Mohammad Qunj who I can't express my feeling in such word that for there suggestion and evolution through out completion my project.

I will never forget them

ABSTRACT

Since the beginning, humankind has sought to use elements in the surrounding environment to make life easier and the tasks at hand more efficient. In keeping with this tradition, people have toyed with and explored the concept of using machines to solve problems since ancient times, Only in this 20th century have significant advances occurred, making the possibility of an actual manifestation of artificial intelligence more and more a reality. The following timeline provides a look at important occurrences in the development of the field of artificial intelligence. Those items in bold print are what we considered the most significant events in the development of AI.

This project presents a study of A.I. system and provides a comparison between them , Neural Network ,Fuzzy system and Expert system are investigated, and a real-life application on A.I. is presented.

TABLE OF CONTEN

ACKNOWLEDGMENT i
ABSTRACT ii
TABLE OF CONTENTS iii
INTRODUCTION 1
CHAPTER ONE: ARTIFICIAL INTELLIGENCE 3
1.1 OVERVIEW 3
1.2. WHAT IS ARTIFICIAL INTELLIGENCE SYSTEM? 3
1.2.1. A GENERAL DEFINITION 4
1.2.2. ANOTHER DEFINITION 4
1.2.3. AN ENGINEERING DEFINITION 4
1.3. ARTIFICIAL INTELLIGENCE HISTORY AND EVOLUTION 6
1.4. THE CHARACTERISTICS OF ARTIFICIAL INTELLIGENCE SYSTEM 10
1.5. ARTIFICIAL INTELLIGENCE METHOD 10
1.5.1. KNOWLEDGE REPRESENTATION 11
1.5.2. SOLUTION SEARCH METHOD 12
1.5.3. METHOD OF REASONING 12
1.5.4. MACHINE LEARNING 13
1.6. ARTIFICIAL INTELLIGENCE PROBLEMS 14
1.7 SUMMARY 14
CHAPTER TWO: EXPERT SYSTEM 16
2.1 OVERVIEW 16
2.2. INTRODUCTION OF EXPERT SYSTEM 16
2.3. WHAT IS EXPERT SYSTEM? 17
2.4. EXPERT SYSTEM DIFINITIONS 18
2.5. DIFINITIONS EXPERT SYSTEM BUILDING TOOLS 19
2.6. EXPERT SYSTEM ARCHITECTURE 20
2.7. PROFILE OF ATOOL: ES/KERNEL2 21
2.7.1. SEMINAL CHARACTERISTICS OF ES/KERNEL2 INCLUDE 22
2.8. RULES AND EXPERT SYSTEM 23
2.9. ADVANTAGES AND DISADVANTAGES OF EXPERT SYSTEM 26
2.9.1. ADVANTAGES OF EXPERT SYSTEMS 26
2.9.2. DISADVANTAGES OF EXPERT SYSTEM 28

2.10. SUMMARY 30

CHAPTER THREE: FUZZY SYSTEM 30
3.1 OVERVIEW 30
3.2. INTRODUCTION TO FUZZY SYSTEM 32
3.3. WHAT IS FUZZY LOGIC? 33
3.3.1 WHAT IS FUZZY LOGIC USED FOR? 33
3.3.2. WHO USES FUZZY LOGIC? 33
3.3.3. WHY IS FUZZY LOGIC BETTER? 33
3.4 FUZZY SETS 35
3.4.1. DEFINITIONS 36
3.4.2 Fuzzy Set Operations 38
3.5. GENERATING FUZZY RULES 40
3.6. EXAMPLE OF FUZZY MODELING 43
3.7. THE FUZZY LOGIC CONTROL PROBLEM 44
3.8. THE FUZZY LOGIC CONTROL DEVELOPMENT 43
3.9 SUMMARY 45
CHAPTER FOUR: NEURAL NETWORKS 46
4.1 OVERVIEW 46
4.2. INTRODUCTION 46
4.3. WHAT IS A NEURAL NETWORKS? 47
4.3.1 SOME DEFINITIONS OF A NEURAL NETWORK 47
4.3.2. WHO IS CONCERNED WITH NEURAL NETWORKS? 48
4.4. HISTORICAL BACKGROUND OF NEURAL NETWORK 48
4.5. HUMAN AND ARTIFICIAL NUEROL 51
4.5.1 HOE THE HUMAN BRAIN LEARNS 51
4.5.2. FROM HUMAN NEURONS TO ARTIFICIAL NEURONS 52
4.6. WHY IT IS DIFFICULT TO MODEL A BRAIN-LIKE NEURAN ETWORK 52
4.7. AN ENGINEERING APPROACH 53
4.7.1 A SIMPLE NEURON 53
4.7.2. FIRING RULES 54
4.7.3. PATTERN RECOGNITION 55
4.7.4. A MORE COMLICATED NEURON 58

4.8. ARCHITECTURE OF NEURAL NETWORK 58

4.8.1. FEED-FORWORD NETWORK 58

4.8.2 FEEDBACK NETWORK 59

4.8.3 NETWORK LAYERS 60

4.9. WHY USE NEURAL NETWORK? 61

4.10. SUMMARY 61

CHAPTER FIVE: A.I .APPLICATIO 63

5.1 OVERVIEW 63

5.2 INTRODUCTION 63

5.3 ARTIFICIAL NEUROL NETWORK 64

5.4 PROBLEM FORMULATION 66

5.5 PROPOSED METHODOLOGY 66

5.5.1 INFORMATION ANALYSIS 68

5.5.2. Model identification 69

5.6. APPLICATIPON TO SAN JUANRIVER BASIN 72

5.7. SUMMARY 76

CONCLUSION 78

REFERENCES 80

INTRODUCTION

The increasing prominence of computers has led to a new way of looking at the world. This view sees nature as a form of computation. By applying Artificial Intelligent (AI) to the computers to solve difficult problems, whose solutions require human intelligence, Together with the neural networks another interesting algorithms approach, inspired by the biological behavior; Artificial Intelligent is being applied in complicated computer systems, along with the neural networks.

The aim of this project is to show and define the Artificial Intelligent system in order to use it wisely and have a look on its problems and their solutions.

In chapter one brief history and background of Artificial Intelligent, Then a discussion of the structure of the Artificial Intelligent

In chapter two brief history of expert system along with biological terminology and their benefits and their structures. A brief discussion of their structures, topologies and their applications.

In chapter three concise fuzzy system. What is the use of fuzzy logic, with some brief discussion of their structures by using soma of fuzzy methods we will see in example?

In chapter four brief history of neural networks along with biological terminology and their benefits and their structures. A brief discussion of their structures, topologies and their applications.

In chapter five the presentation of the applications in the artificial intelligence. Dealing with examples related to applying the artificial intelligence, monthly stream flow prediction using artificial neural networks (ANN) on mountain watersheds. The

procedure addresses the Selection of input variables, the definition of model architecture and the strategy of the learning process.

CHAPTER ONE

Artificial Intelligence system

1.1. Overview

This chapter is an introduction to Artificial Intelligence, Artificial intelligence or AI shortly is a relative young discipline compared to the by now mature nature sciences such as physical or chemistry or biology and mathematics. Detailed historical background is provided. A method to choose responses according to its objectives and memories, and some way of performing these responses in and on its environment.

1.2. What is Artificial Intelligence System?

Of fundamental concern in the application of artificial intelligence is the question "what is artificial intelligence?" and providing a straightforward.

The term Artificial is perhaps simple enough to understand, this meaning contrived, synthetic, man-made, but what is intelligence? We don't really know intelligence means; the field of artificial intelligence has been in existence for approximately 40 years and provides us with a working and tracking it.

The majority of attempts to precisely define the many-faceted term artificial intelligence do not completely succeed the failure here is due to the

- Non-existence of a precise and comprehensive definition (natural) intelligence itself
- Scope and depth of artificial intelligence in terms of its wide application area and the extent of problem to be solved

  And some sources define intelligence as the
- Ability to acquire, analyze, understand and creatively apply the knowledge

- Ability to reason (think) and intelligently handle (behave)

1.2.1. A General Definition

Widely accepted definition of artificial intelligence are both controversial and elusive considering the difficulty in defining natural intelligence (NT), it is probable be better then to attempt definition of AI then the origin of AI may be traced back to conference at Dartmouth college in the summer of 1956 and the broadest definition is that:

AI is field of study that seeks to explain and emulate intelligent behavior in term of computational processes

1.2.2. Another Definition

- Artificial intelligence is activity carried out by machine that, if carried by human, would be considered intelligent. From practical point of view, simulating intelligence is just a good as actual intelligence
- Artificial intelligence is branch of computer science dealing with computer system implementing a restricted but definite part of human intelligence, particularly in knowledge acquisition, perception learning, reasoning, language and scan understanding.
- AI is the field of study that seeks to explain and emulate intelligent behavior in the terms of computational processes.

1.2.3. An Engineering Definition

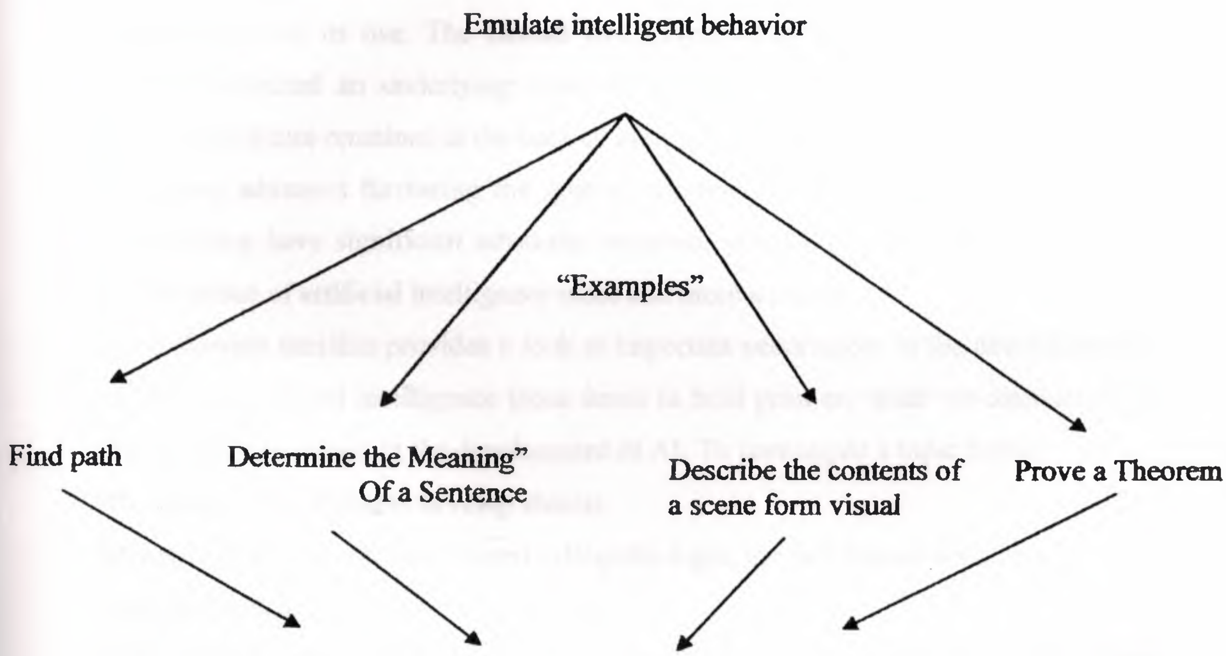
Since the applied artificial intelligence, concerned with implementation of intelligence system, is type of engineering, or at least an applied science, the question is that whether artificial intelligence itself is consequently a branch of engineering or an application field of cognitive science.

From an engineering viewpoint that artificial intelligence is about:

"Generating representation and procedures that automatically solve problem heretofore solved by humans"

An engineering approach to AI requires the development of programs; that is algorithms and databases that exhibit intelligent behavior. Figure 1.1 shows characterization of intelligent behavior without defining intelligence. Since this autonomous capability is a form of advanced computation, an alternative descriptor might be machine intelligence this reinforces our previous definition since:

- Mechanization of intelligence implies the need for an explicit and quantitative description
- Codifying expert knowledge is articulating intelligence.



These are the ramification of intelligence behavior that do not require general definition or characterization of intelligence.

Figure 1.1 AI goals via example of intelligence behavior

1.3. Artificial Intelligence History and Evolution

Since the beginning, humankind has sought to use elements in the surrounding environment to make life easier and the tasks at hand more efficient. In keeping with this tradition, people have toyed with and explored the concept of using machines to solve problems since ancient times. The development of these ideas can be seen as far back as ancient Greece, with the mention of intelligent machines in mythology (e.g., Ephesus and Pygmalion). Most people are aware of the development of calculators ("the brains of AI") throughout history. The earliest type was the abacus, which was used in China. The Egyptians invented a counting machine that used pebbles some time before Herodotus noted its use. The Greeks and Romans had similar devices. These early attempts reflected an underlying desire to replicate human reasoning in nonhuman forms. This desire remained at the back of human consciousness over the centuries, with occasional advances furthering the goal of creating 'thinking machines.' Only in this 20th century have significant advances occurred, making the possibility of an actual manifestation of artificial intelligence more and more a reality.[1]

The following timeline provides a look at important occurrences in the development of the field of artificial intelligence those items in bold print are what we considered the most significant events in the development of AI. To investigate a topic further

6th century B.C. Chinese develop abacus

5th century B.C. Aristotle invented syllogistic logic, the first formal deductive reasoning system.

13th century Talking heads were said to have been created, Roger Bacon and Albert the Great reputedly among the owners. Ramon Lull, Spanish theologian, invented machines for discovering nonmathematical truths through combinatory

15th century Invention of printing using moveable type. Gutenberg Bible printed (1456).

15th-16th century Clocks, the first modern measuring machines, were first produced using lathes.

16th century Clockmakers extended their craft to creating mechanical animals and other novelties. Rabbi Loews of Prague is said to have invented the Golem, a clay man brought to life (1580).

17th century early in the century, Descartes proposed that bodies of animals are nothing more than complex machines. Many other 17th century thinkers offered variations and elaborations of Cartesian mechanism Wilhelm Schickard (1592-1635), invented an automatic digital calculator (1633) Hobbes published The Leviathan, containing a material and combinatorial theory of thinking. Pascal created the first mechanical digital calculating machine (1642). Leibniz improved Pascal's machine to do multiplication & division (1673) and envisioned a universal calculus of reasoning by which argument could be decided mechanically

18th century the 18th century saw a profusion of mechanical toys, including the celebrated mechanical duck of Vaucanson and von Kemp Len's phony mechanical chessplayer,TheTurk(1769)

19th century Ladies (led by Ned Ladd) destroyed machinery in England (1811- 1816). Mary Shelley published the story of Frankenstein's monster (1818). George Boole developed a binary algebra representing (some) "laws of thought." **20th century** Bertrand Russell and Alfred North Whitehead published Principia Mathematic, which revolutionized formal logic. Russell, Ludwig Wittgenstein, and Rudolf Carnal lead philosophy into logical analysis of knowledge, as it shown in Figure 1.2 timeline provides a look at important occurrences in the development of the field of artificial intelligence.[2]

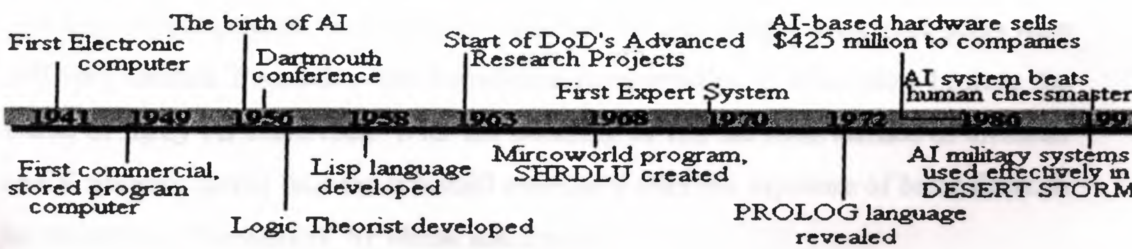


Figure 1.2 AI timeline provide

The Birth of AI (1945-56):

It was the postwar period (1945-1956) that Artificial Intelligence first emerged as a widely discussed field. What propelled the birth of Artificial Intelligence was the arrival of modern computer technology. The development of the modern computer technology affected the AI research tremendously. Many pioneers of AI broke away from the traditional approach of artificial neurons and decided that the human thought could be more efficiently emulated with modern digital computer. Those who did not accept digital computers as the new approach stayed in the parallel field of neural network [3]

The Dawning Age of AI (1956-63)

1956-1963 represents the dawning of an intensive AI wave. During this period, major AI research centers concentrated their work on two main themes. First, the attempt to limit the breadth of searches in trial-and-error problems led to the initiation of projects such as Logic Theorist (considered as the first AI program), Geometry Theorem Proved, and SAINT. Next, the study on computer learning includes projects on chess, checkers, and pattern recognition programs. Specialized list-processing AI languages such as LISP were also developed in MIT and other places in 1958.[4]

The Maturation of AI (1963-70)

By mid 60's, AI had become the common goal of thousands of different studies. AI researchers utilized their programming techniques and the improved computers in pursuing various projects. However, the memories of computers during these years were still very limited. Perception and knowledge representation in computers became the theme of many AI researches. With the booming of AI, the rival science of artificial neural network would face the downfall especially after the exposure of basic flaws in its researching "Perception" by Minsk and Popert.

The Specialization of Various AI Studies (1970's)

Different AI-related studies had developed into recognizable specialties during the 70's. Edward Feigenbaum pioneered the research on expert systems; Roger Shank promoted language analysis with a new way of interpreting the meaning of words; Marvin Minks propelled the field of knowledge representation a step further with his new structures for

representing mental constructs; Douglas Lenat explored automatic learning and the nature of heuristics; David Marr improved computer vision; the authors of PROLOG language presented a convenient higher language for AI researches. The specialization of AI in the 70's greatly strengthened the backbone of AI theories. However, AI application were still few and premature.[5]

The Unfulfilled Expectations (1980's)

The 1980's were a period of roller coasting for AI. The anti-science tradition of the public was improved greatly following the appearance of Star Wars movies and the new popularity of the personal computers. XCON, the first expert system employed in industrial world, symbolized the budding of real AI application. Within four years, XCON had grown tenfold with an investment of fifty person-years in the program and an achievement of saving about forty million dollar's in testing and manufacturing costs for the industrial clients. Following the brilliant success was the AI boom. The number of AI groups increased tremendously and in 1985, 150 companies spent about $1 billion altogether on internal AI groups. However, the fundamental AI algorithm was still unsatisfying. As Marvin Minsk warned the over-confident public: these seemingly intelligent programs simply make dumb decisions faster. Indeed, the warning foreshadowed the downfall of AI industry in late 80's. The replacing of LISP machines by standard microcomputers with AI software's in the popular C language in 1987 and the instability of expert systems caused a painful transition on expert system industry; the computer vision industry also suffered from a big setback when Machine Vision International crashed in 1988; one other major loss was the failure in Autonomous Land Vehicle project (AI drivers + Robotics). The AI industry started recovering at the end of the 80's but learning from the past experience, public assumed a much more conservative view on AI ever since. Another notable event is the revisiting of neural network with the work done by the Parallel Distributed Processing Study Group. In 1989, about three hundred companies were founded to compete for the predicted $1 billion market for neural nets by end of the century. [6]

AI Being Incorporated in War (early 1990's)

The Persian Gulf War in the early 90's proved the importance of AI research for military use. Tasks as simple as packing a transport plane and as complicated as the timing and coordination of Operation Desert Storm were assisted by AI-oriented expert systems. Advanced weapons such as "cruise missiles" were equipped with technologies previously studies in different AI-related fields such as Robotics and Machine Vision. Two projects succession the Automated Land Vehicle project were the Pilot's Associate project (electronic copilot) and the Battle Management System project (military expert systems).[7]

New AI Applications (late 1990's)

The victory of Deep Blue over chess champion Kasparov in 1996 led to a new summit of AI gaming. A new branch of expert systems has been expected to prosper as Genetic Engineering matures. Manipulating such gigantic knowledge base of human DNA map (Bioinformatics) will require very specialized algorithms and AI researches.[8]

1.4. The Characteristics of Artificial Intelligence System

- Imitation of the human reasoning process.
- Sequential information processing.
- Explicit knowledge representation.
- Use of deductive reasoning.
- Learning is outside system.

1.5. Artificial Intelligence Method

Artificial intelligence problems span abroad array of application areas of human active. The problem to be solved are not only numerous but also divers, and quit dissimilar, they require fundamentally different approaches for problem solution. But the question that consequently arises is whether there are same common, general methods of problem solving, compatible with the large spectrum of solution? And of course the answer is a great extent affirmative, the most suitable AI methods will be

briefly reviewed, in terms of the role they play in the process of problem of solving, the method are grouped into:

- Knowledge representation
- Solution search
- Reasoning
- Machine learning

1.5.1. Knowledge representation

Knowledge representations reasoning with the knowledge are two major building blocks of contemporary AI system, capturing for later essential features of a knowledge-domain in a form convenient for later knowledge processing is the first constructive step towards the building of an intelligent, Knowledge-based system belonging to the knowledge acquisition phase of the building process. Here a form has to be found for the abstract representation of facts and the relationship between the facts that will cover as much of domain knowledge as possible. In addition to confined generality, knowledge representation method should include the representation of qualitative and semantic knowledge as well as meat- knowledge with reference to this possible knowledge levels to dealt in the AI as depicted in fig1.3 should be kept in mind.

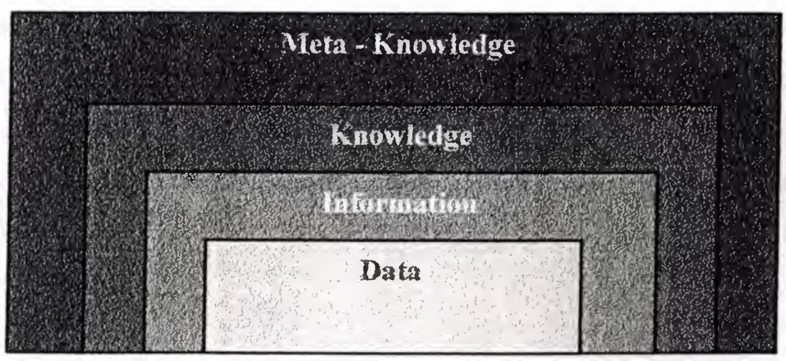


Figure 1.3 Knowledge Levels

1.5.2. Solution search method

The pivotal issues of problem solving strategies are classified to be aimed at goal finding, but before a problem can be solved it has to be exactly defined first, consequently the intrinsic problem solving method includes a:

- Problem representation
- Solution search

Method "problem representation is prerequisite for a solution search to be applied, prior to the application we must know where the total problem is allocated where we are at the beginning of the problem solution where the goal to be pursued is within the whole constellation and how to reach it" In AI a widely accepted idea that came out of the early research is that the most adequate problem formulation is its state-space representation in term of

- Initial state, from which the solutions search start.
- Terminal state i.e. the state that represents the problem solution or its goal.
- Operation i.e. starts transformation to be employed for stepwise move from the initial to the goal start.

1.5.3. Method of Reasoning

The next major group of AI is centered on the problem of Reasoning with the stored knowledge, Reasoning is actually drawing conclusion from the facts or actually inferring conclusions from premises, the method however aside from the way the knowledge engineer or the domain expert uses it for his purposes can be reduced mainly to the representation forms understanding by computer of which the following three are basic, namely

- Logic expressions
- Production rules
- Slot-and-filler structures

The most natural approach in developing reasoning method would be to first study that what is known as common sense reasoning based on common sense knowledge this is what each person starting any age can do and does: he reasons about the space and abject of his surrounding, about their shape, colors, and dimension, about the time, event and the sequence of event etc, there is same serious critical option that we may not be able build an intelligent program that will be superior to a 3 year old child ,the state-of-the -art here is that AI although able to solve domain expert problem.

On other hand, the method of reasoning of called automated reasoning has been much more successful and is used in expert system it is based on logic programming in which reasoning is already built in defined in terms of mathematical logic, automated is reasoning is a process of using some unambiguous notations for representing knowledge in order to draw corresponding inference. In diagnostic expert system there are two levels of knowledge representation of the system to be identified:

- "Shallow" knowledge representation, description the system under diagnosis by set of heuristic chunks of established facts.
- "Deep" knowledge representation that includes the description of the structure and the function behavior of the system under diagnosis.

1.5.4. Machine learning

An exact definition of learning is difficult to find, it might be any change to an intelligent system, such as the addition of any single facts, implanting of a new piece of knowledge or a control strategy, Simon (1983) says that the system changes in the process of learning directly contribute to the improvement of its efficiency in the sense of its better behavior when solving more of the same or similar problem, this imposition resembles the definition of learning in psychology, where learning is any change in the subject's behavior to repeated situation, after reducing other factory . Automated learning should be defined as the capability of an intelligent system to machine improve its behavior (or its performance) as a result of its previous experience, two outstanding of automated learning are:

- Concept learning
- Inductive learning, or learning by example

NEAR EAST UNIVERSITY

NEAR EAST UNIVERSITY
LIBRARY
1988 - LEFKOŞA

Faculty of Engineering

Department of Computer Engineering

ARTIFICIAL INTELLIGENCE SYSTEM

Graduation Project

COM – 400

Student: Nader Ibraik

Supervisor: Assoc. Prof. Dr Adnan Khashman

Nicosia - 2003

ACKNOWLEDGEMENTS

First of all I am thanking full to the most gracious "ALLAH" the almighty, who enable me to complete this project

Secondly, I would like to award my supervisor Assoc. Prof. Dr Adnan Khashman for being so operative averring supervise me in this work, and for his overwhelming and limitless help he had done to me.

Thirdly, I will never over look the encourage I had resaved from all my family, specially my parents, my "father" the best father in this world, and the best sweetest woman in the word "mother", for there supporting me and caring for me, I am thankful for them and my family members from brother and sisters.

Fourthly this will be the most important moment in my life that I will settlement my project to the best brother" ZAKY-IBRAIK" because of his encourage me and supporting me in the long of my 4th year study ,I am so thankful for him, my brother I am own you too match.

Finally I like to thank the best friends I had met them in Cyprus Mohammed Asfour and Mohammad Shuqair and Mohammad Qunj who I can't express my feeling in such word that for there suggestion and evolution through out completion my project.

I will never forget them

ABSTRACT

Since the beginning, humankind has sought to use elements in the surrounding environment to make life easier and the tasks at hand more efficient. In keeping with this tradition, people have toyed with and explored the concept of using machines to solve problems since ancient times, Only in this 20th century have significant advances occurred, making the possibility of an actual manifestation of artificial intelligence more and more a reality. The following timeline provides a look at important occurrences in the development of the field of artificial intelligence. Those items in bold print are what we considered the most significant events in the development of AI.

This project presents a study of A.I. system and provides a comparison between them , Neural Network ,Fuzzy system and Expert system are investigated, and a real-life application on A.I. is presented.

TABLE OF CONTEN

ACKNOWLEDGMENT	i
ABSTRACT	ii
TABLE OF CONTENTS	iii
INTRODUCTION	1
CHAPTER ONE: ARTIFICIAL INTELLIGENCE	3
1.1 OVERVIEW	3
1.2. WHAT IS ARTIFICIAL INTELLIGENCE SYSTEM?	3
1.2.1. A GENERAL DEFINITION	4
1.2.2. ANOTHER DEFINITION	4
1.2.3. AN ENGINEERING DEFINITION	4
1.3. ARTIFICIAL INTELLIGENCE HISTORY AND EVOLUTION	6
1.4. THE CHARACTERISTICS OF ARTIFICIAL INTELLIGENCE SYSTEM	10
1.5. ARTIFICIAL INTELLIGENCE METHOD	10
1.5.1. KNOWLEDGE REPRESENTATION	11
1.5.2. SOLUTION SEARCH METHOD	12
1.5.3. METHOD OF REASONING	12
1.5.4. MACHINE LEARNING	13
1.6. ARTIFICIAL INTELLIGENCE PROBLEMS	14
1.7 SUMMARY	14
CHAPTER TWO: EXPERT SYSTEM	16
2.1 OVERVIEW	16
2.2. INTRODUCTION OF EXPERT SYSTEM	16
2.3. WHAT IS EXPERT SYSTEM?	17
2.4. EXPERT SYSTEM DIFINITIONS	18
2.5. DIFINITIONS EXPERT SYSTEM BUILDING TOOLS	19
2.6. EXPERT SYSTEM ARCHITECTURE	20
2.7. PROFILE OF ATOOL: ES/KERNEL2	21
2.7.1. SEMINAL CHARACTERISTICS OF ES/KERNEL2 INCLUDE	22
2.8. RULES AND EXPERT SYSTEM	23
2.9. ADVANTAGES AND DISADVANTAGES OF EXPERT SYSTEM	26
2.9.1. ADVANTAGES OF EXPERT SYSTEMS	26
2.9.2. DISADVANTAGES OF EXPERT SYSTEM	28

2.10. SUMMARY 30

CHAPTER THREE: FUZZY SYSTEM 30
3.1 OVERVIEW 30
3.2. INTRODUCTION TO FUZZY SYSTEM 32
3.3. WHAT IS FUZZY LOGIC? 33
3.3.1 WHAT IS FUZZY LOGIC USED FOR? 33
3.3.2. WHO USES FUZZY LOGIC? 33
3.3.3. WHY IS FUZZY LOGIC BETTER? 33
3.4 FUZZY SETS 35
3.4.1. DEFINITIONS 36
3.4.2 Fuzzy Set Operations 38
3.5. GENERATING FUZZY RULES 40
3.6. EXAMPLE OF FUZZY MODELING 43
3.7. THE FUZZY LOGIC CONTROL PROBLEM 44
3.8. THE FUZZY LOGIC CONTROL DEVELOPMENT 43
3.9 SUMMARY 45
CHAPTER FOUR: NEURAL NETWORKS 46
4.1 OVERVIEW 46
4.2. INTRODUCTION 46
4.3. WHAT IS A NEURAL NETWORKS? 47
4.3.1 SOME DEFINITIONS OF A NEURAL NETWORK 47
4.3.2. WHO IS CONCERNED WITH NEURAL NETWORKS? 48
4.4. HISTORICAL BACKGROUND OF NEURAL NETWORK 48
4.5. HUMAN AND ARTIFICIAL NUEROL 51
4.5.1 HOE THE HUMAN BRAIN LEARNS 51
4.5.2. FROM HUMAN NEURONS TO ARTIFICIAL NEURONS 52
4.6. WHY IT IS DIFFICULT TO MODEL A BRAIN-LIKE NEURAN ETWORK 52
4.7. AN ENGINEERING APPROACH 53
4.7.1 A SIMPLE NEURON 53
4.7.2. FIRING RULES 54
4.7.3. PATTERN RECOGNITION 55
4.7.4. A MORE COMLICATED NEURON 58

4.8. ARCHITECTURE OF NEURAL NETWORK 58

4.8.1. FEED-FORWORD NETWORK 58

4.8.2 FEEDBACK NETWORK 59

4.8.3 NETWORK LAYERS 60

4.9. WHY USE NEURAL NETWORK? 61

4.10. SUMMARY 61

CHAPTER FIVE: A.I .APPLICATIO 63

5.1 OVERVIEW 63

5.2 INTRODUCTION 63

5.3 ARTIFICIAL NEUROL NETWORK 64

5.4 PROBLEM FORMULATION 66

5.5 PROPOSED METHODOLOGY 66

5.5.1 INFORMATION ANALYSIS 68

5.5.2. Model identification 69

5.6. APPLICATIPON TO SAN JUANRIVER BASIN 72

5.7. SUMMARY 76

CONCLUSION 78

REFERENCES 80

INTRODUCTION

The increasing prominence of computers has led to a new way of looking at the world. This view sees nature as a form of computation. By applying Artificial Intelligent (AI) to the computers to solve difficult problems, whose solutions require human intelligence, Together with the neural networks another interesting algorithms approach, inspired by the biological behavior; Artificial Intelligent is being applied in complicated computer systems, along with the neural networks.

The aim of this project is to show and define the Artificial Intelligent system in order to use it wisely and have a look on its problems and their solutions.

In chapter one brief history and background of Artificial Intelligent, Then a discussion of the structure of the Artificial Intelligent

In chapter two brief history of expert system along with biological terminology and their benefits and their structures. A brief discussion of their structures, topologies and their applications.

In chapter three concise fuzzy system. What is the use of fuzzy logic, with some brief discussion of their structures by using soma of fuzzy methods we will see in example?

In chapter four brief history of neural networks along with biological terminology and their benefits and their structures. A brief discussion of their structures, topologies and their applications.

In chapter five the presentation of the applications in the artificial intelligence. Dealing with examples related to applying the artificial intelligence, monthly stream flow prediction using artificial neural networks (ANN) on mountain watersheds. The

procedure addresses the Selection of input variables, the definition of model architecture and the strategy of the learning process.

CHAPTER ONE

Artificial Intelligence system

1.1. Overview

This chapter is an introduction to Artificial Intelligence, Artificial intelligence or AI shortly is a relative young discipline compared to the by now mature nature sciences such as physical or chemistry or biology and mathematics. Detailed historical background is provided. A method to choose responses according to its objectives and memories, and some way of performing these responses in and on its environment.

1.2. What is Artificial Intelligence System?

Of fundamental concern in the application of artificial intelligence is the question "what is artificial intelligence?" and providing a straightforward.

The term Artificial is perhaps simple enough to understand, this meaning contrived, synthetic, man-made, but what is intelligence? We don't really know intelligence means; the field of artificial intelligence has been in existence for approximately 40 years and provides us with a working and tracking it.

The majority of attempts to precisely define the many-faceted term artificial intelligence do not completely succeed the failure here is due to the

- Non-existence of a precise and comprehensive definition (natural) intelligence itself
- Scope and depth of artificial intelligence in terms of its wide application area and the extent of problem to be solved

  And some sources define intelligence as the
- Ability to acquire, analyze, understand and creatively apply the knowledge

- Ability to reason (think) and intelligently handle (behave)

1.2.1. A General Definition

Widely accepted definition of artificial intelligence are both controversial and elusive considering the difficulty in defining natural intelligence (NT), it is probable be better then to attempt definition of AI then the origin of AI may be traced back to conference at Dartmouth college in the summer of 1956 and the broadest definition is that:

AI is field of study that seeks to explain and emulate intelligent behavior in term of computational processes

1.2.2. Another Definition

- Artificial intelligence is activity carried out by machine that, if carried by human, would be considered intelligent. From practical point of view, simulating intelligence is just a good as actual intelligence
- Artificial intelligence is branch of computer science dealing with computer system implementing a restricted but definite part of human intelligence, particularly in knowledge acquisition, perception learning, reasoning, language and scan understanding.
- AI is the field of study that seeks to explain and emulate intelligent behavior in the terms of computational processes.

1.2.3. An Engineering Definition

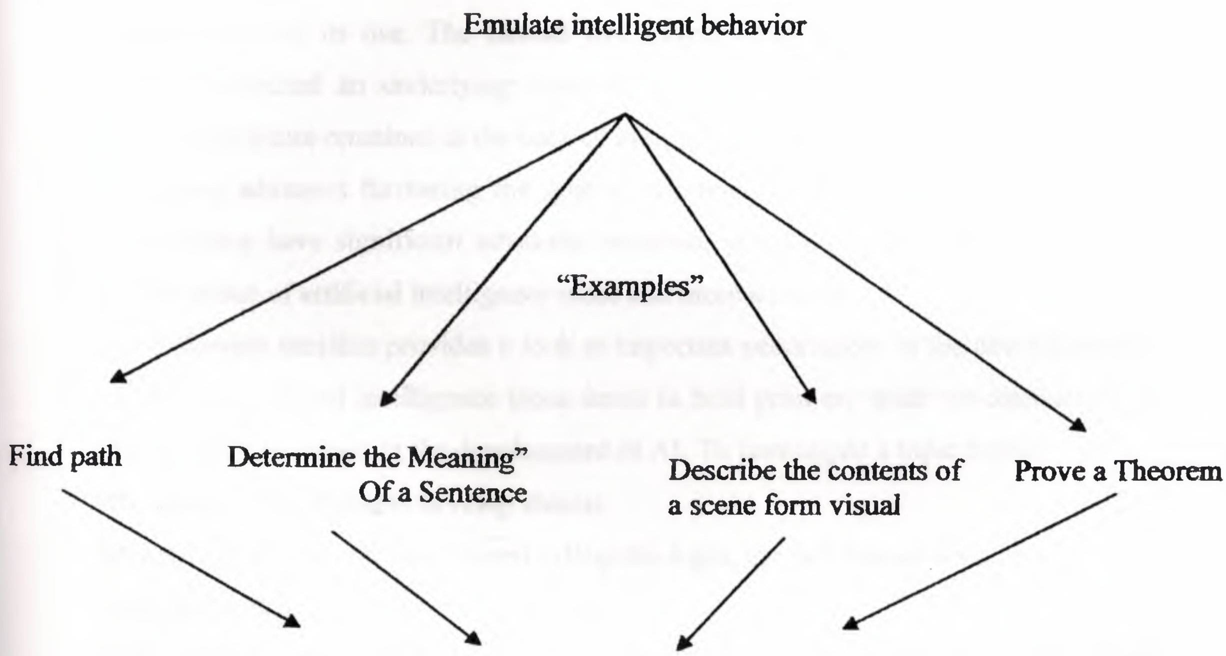
Since the applied artificial intelligence, concerned with implementation of intelligence system, is type of engineering, or at least an applied science, the question is that whether artificial intelligence itself is consequently a branch of engineering or an application field of cognitive science.

From an engineering viewpoint that artificial intelligence is about:

"Generating representation and procedures that automatically solve problem heretofore solved by humans"

An engineering approach to AI requires the development of programs; that is algorithms and databases that exhibit intelligent behavior. Figure 1.1 shows characterization of intelligent behavior without defining intelligence. Since this autonomous capability is a form of advanced computation, an alternative descriptor might be machine intelligence this reinforces our previous definition since:

- Mechanization of intelligence implies the need for an explicit and quantitative description
- Codifying expert knowledge is articulating intelligence.



These are the ramification of intelligence behavior that do not require general definition or characterization of intelligence.

Figure 1.1 AI goals via example of intelligence behavior

1.3. Artificial Intelligence History and Evolution

Since the beginning, humankind has sought to use elements in the surrounding environment to make life easier and the tasks at hand more efficient. In keeping with this tradition, people have toyed with and explored the concept of using machines to solve problems since ancient times. The development of these ideas can be seen as far back as ancient Greece, with the mention of intelligent machines in mythology (e.g., Ephesus and Pygmalion). Most people are aware of the development of calculators ("the brains of AI") throughout history. The earliest type was the abacus, which was used in China. The Egyptians invented a counting machine that used pebbles some time before Herodotus noted its use. The Greeks and Romans had similar devices. These early attempts reflected an underlying desire to replicate human reasoning in nonhuman forms. This desire remained at the back of human consciousness over the centuries, with occasional advances furthering the goal of creating 'thinking machines.' Only in this 20th century have significant advances occurred, making the possibility of an actual manifestation of artificial intelligence more and more a reality.[1]

The following timeline provides a look at important occurrences in the development of the field of artificial intelligence those items in bold print are what we considered the most significant events in the development of AI. To investigate a topic further

6th century B.C. Chinese develop abacus

5th century B.C. Aristotle invented syllogistic logic, the first formal deductive reasoning system.

13th century Talking heads were said to have been created, Roger Bacon and Albert the Great reputedly among the owners. Ramon Lull, Spanish theologian, invented machines for discovering nonmathematical truths through combinatory

15th century Invention of printing using moveable type. Gutenberg Bible printed (1456).

15th-16th century Clocks, the first modern measuring machines, were first produced using lathes.

16th century Clockmakers extended their craft to creating mechanical animals and other novelties. Rabbi Loews of Prague is said to have invented the Golem, a clay man brought to life (1580).

17th century early in the century, Descartes proposed that bodies of animals are nothing more than complex machines. Many other 17th century thinkers offered variations and elaborations of Cartesian mechanism Wilhelm Schickard (1592-1635), invented an automatic digital calculator (1633) Hobbes published The Leviathan, containing a material and combinatorial theory of thinking. Pascal created the first mechanical digital calculating machine (1642). Leibniz improved Pascal's machine to do multiplication & division (1673) and envisioned a universal calculus of reasoning by which argument could be decided mechanically

18th century the 18th century saw a profusion of mechanical toys, including the celebrated mechanical duck of Vaucanson and von Kemp Len's phony mechanical chessplayer,TheTurk(1769)

19th century Ladies (led by Ned Ladd) destroyed machinery in England (1811- 1816). Mary Shelley published the story of Frankenstein's monster (1818). George Boole developed a binary algebra representing (some) "laws of thought."

20th century Bertrand Russell and Alfred North Whitehead published Principia Mathematic, which revolutionized formal logic. Russell, Ludwig Wittgenstein, and Rudolf Carnal lead philosophy into logical analysis of knowledge, as it shown in Figure 1.2 timeline provides a look at important occurrences in the development of the field of artificial intelligence.[2]

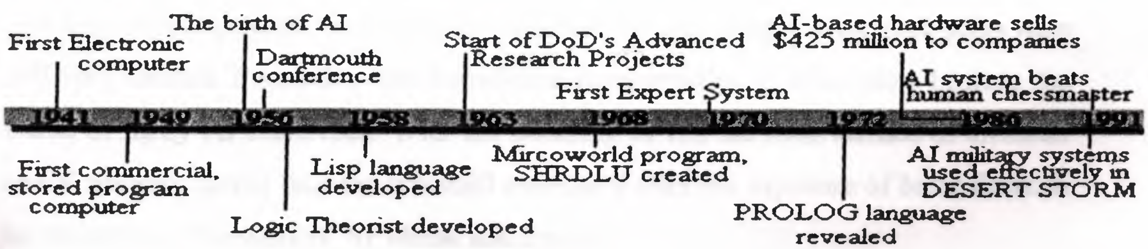


Figure 1.2 AI timeline provide

The Birth of AI (1945-56):

It was the postwar period (1945-1956) that Artificial Intelligence first emerged as a widely discussed field. What propelled the birth of Artificial Intelligence was the arrival of modern computer technology. The development of the modern computer technology affected the AI research tremendously. Many pioneers of AI broke away from the traditional approach of artificial neurons and decided that the human thought could be more efficiently emulated with modern digital computer. Those who did not accept digital computers as the new approach stayed in the parallel field of neural network [3]

The Dawning Age of AI (1956-63)

1956-1963 represents the dawning of an intensive AI wave. During this period, major AI research centers concentrated their work on two main themes. First, the attempt to limit the breadth of searches in trial-and-error problems led to the initiation of projects such as Logic Theorist (considered as the first AI program), Geometry Theorem Proved, and SAINT. Next, the study on computer learning includes projects on chess, checkers, and pattern recognition programs. Specialized list-processing AI languages such as LISP were also developed in MIT and other places in 1958.[4]

The Maturation of AI (1963-70)

By mid 60's, AI had become the common goal of thousands of different studies. AI researchers utilized their programming techniques and the improved computers in pursuing various projects. However, the memories of computers during these years were still very limited. Perception and knowledge representation in computers became the theme of many AI researches. With the booming of AI, the rival science of artificial neural network would face the downfall especially after the exposure of basic flaws in its researching "Perception" by Minsk and Popert.

The Specialization of Various AI Studies (1970's)

Different AI-related studies had developed into recognizable specialties during the 70's. Edward Feigenbaum pioneered the research on expert systems; Roger Shank promoted language analysis with a new way of interpreting the meaning of words; Marvin Minks propelled the field of knowledge representation a step further with his new structures for

representing mental constructs; Douglas Lenat explored automatic learning and the nature of heuristics; David Marr improved computer vision; the authors of PROLOG language presented a convenient higher language for AI researches. The specialization of AI in the 70's greatly strengthened the backbone of AI theories. However, AI application were still few and premature.[5]

The Unfulfilled Expectations (1980's)

The 1980's were a period of roller coasting for AI. The anti-science tradition of the public was improved greatly following the appearance of Star Wars movies and the new popularity of the personal computers. XCON, the first expert system employed in industrial world, symbolized the budding of real AI application. Within four years, XCON had grown tenfold with an investment of fifty person-years in the program and an achievement of saving about forty million dollar's in testing and manufacturing costs for the industrial clients. Following the brilliant success was the AI boom. The number of AI groups increased tremendously and in 1985, 150 companies spent about $1 billion altogether on internal AI groups. However, the fundamental AI algorithm was still unsatisfying. As Marvin Minsk warned the over-confident public: these seemingly intelligent programs simply make dumb decisions faster. Indeed, the warning foreshadowed the downfall of AI industry in late 80's. The replacing of LISP machines by standard microcomputers with AI software's in the popular C language in 1987 and the instability of expert systems caused a painful transition on expert system industry; the computer vision industry also suffered from a big setback when Machine Vision International crashed in 1988; one other major loss was the failure in Autonomous Land Vehicle project (AI drivers + Robotics). The AI industry started recovering at the end of the 80's but learning from the past experience, public assumed a much more conservative view on AI ever since. Another notable event is the revisiting of neural network with the work done by the Parallel Distributed Processing Study Group. In 1989, about three hundred companies were founded to compete for the predicted $1 billion market for neural nets by end of the century. [6]

AI Being Incorporated in War (early 1990's)

The Persian Gulf War in the early 90's proved the importance of AI research for military use. Tasks as simple as packing a transport plane and as complicated as the timing and coordination of Operation Desert Storm were assisted by AI-oriented expert systems. Advanced weapons such as "cruise missiles" were equipped with technologies previously studies in different AI-related fields such as Robotics and Machine Vision. Two projects succession the Automated Land Vehicle project were the Pilot's Associate project (electronic copilot) and the Battle Management System project (military expert systems).[7]

New AI Applications (late 1990's)

The victory of Deep Blue over chess champion Kasparov in 1996 led to a new summit of AI gaming. A new branch of expert systems has been expected to prosper as Genetic Engineering matures. Manipulating such gigantic knowledge base of human DNA map (Bioinformatics) will require very specialized algorithms and AI researches.[8]

1.4. The Characteristics of Artificial Intelligence System

- Imitation of the human reasoning process.
- Sequential information processing.
- Explicit knowledge representation.
- Use of deductive reasoning.
- Learning is outside system.

1.5. Artificial Intelligence Method

Artificial intelligence problems span abroad array of application areas of human active. The problem to be solved are not only numerous but also divers, and quit dissimilar, they require fundamentally different approaches for problem solution. But the question that consequently arises is whether there are same common, general methods of problem solving, compatible with the large spectrum of solution? And of course the answer is a great extent affirmative, the most suitable AI methods will be

briefly reviewed, in terms of the role they play in the process of problem of solving, the method are grouped into:

- Knowledge representation
- Solution search
- Reasoning
- Machine learning

1.5.1. Knowledge representation

Knowledge representations reasoning with the knowledge are two major building blocks of contemporary AI system, capturing for later essential features of a knowledge-domain in a form convenient for later knowledge processing is the first constructive step towards the building of an intelligent, Knowledge-based system belonging to the knowledge acquisition phase of the building process. Here a form has to be found for the abstract representation of facts and the relationship between the facts that will cover as much of domain knowledge as possible. In addition to confined generality, knowledge representation method should include the representation of qualitative and semantic knowledge as well as meat- knowledge with reference to this possible knowledge levels to dealt in the AI as depicted in fig1.3 should be kept in mind.

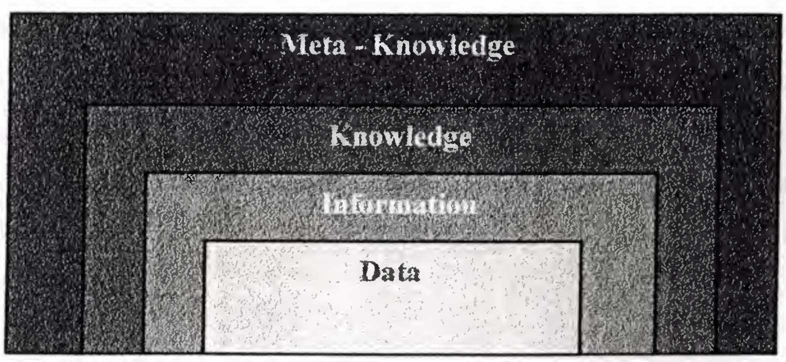


Figure 1.3 Knowledge Levels

1.5.2. Solution search method

The pivotal issues of problem solving strategies are classified to be aimed at goal finding, but before a problem can be solved it has to be exactly defined first, consequently the intrinsic problem solving method includes a:

- Problem representation
- Solution search

Method "problem representation is prerequisite for a solution search to be applied, prior to the application we must know where the total problem is allocated where we are at the beginning of the problem solution where the goal to be pursued is within the whole constellation and how to reach it" In AI a widely accepted idea that came out of the early research is that the most adequate problem formulation is its state-space representation in term of

- Initial state, from which the solutions search start.
- Terminal state i.e. the state that represents the problem solution or its goal.
- Operation i.e. starts transformation to be employed for stepwise move from the initial to the goal start.

1.5.3. Method of Reasoning

The next major group of AI is centered on the problem of Reasoning with the stored knowledge, Reasoning is actually drawing conclusion from the facts or actually inferring conclusions from premises, the method however aside from the way the knowledge engineer or the domain expert uses it for his purposes can be reduced mainly to the representation forms understanding by computer of which the following three are basic, namely

- Logic expressions
- Production rules
- Slot-and-filler structures

The most natural approach in developing reasoning method would be to first study that what is known as common sense reasoning based on common sense knowledge this is what each person starting any age can do and does: he reasons about the space and abject of his surrounding, about their shape, colors, and dimension, about the time, event and the sequence of event etc, there is same serious critical option that we may not be able build an intelligent program that will be superior to a 3 year old child ,the state-of-the –art here is that AI although able to solve domain expert problem.

On other hand, the method of reasoning of called automated reasoning has been much more successful and is used in expert system it is based on logic programming in which reasoning is already built in defined in terms of mathematical logic, automated is reasoning is a process of using some unambiguous notations for representing knowledge in order to draw corresponding inference. In diagnostic expert system there are two levels of knowledge representation of the system to be identified:

- "Shallow" knowledge representation, description the system under diagnosis by set of heuristic chunks of established facts.
- "Deep" knowledge representation that includes the description of the structure and the function behavior of the system under diagnosis.

1.5.4. Machine learning

An exact definition of learning is difficult to find, it might be any change to an intelligent system, such as the addition of any single facts, implanting of a new piece of knowledge or a control strategy, Simon (1983) says that the system changes in the process of learning directly contribute to the improvement of its efficiency in the sense of its better behavior when solving more of the same or similar problem, this imposition resembles the definition of learning in psychology, where learning is any change in the subject's behavior to repeated situation, after reducing other factory . Automated learning should be defined as the capability of an intelligent system to machine improve its behavior (or its performance) as a result of its previous experience, two outstanding of automated learning are:

- Concept learning
- Inductive learning, or learning by example

1.6. Artificial Intelligence Problems

The most natural definition of AI problem would be that the fundamental AI problem is problem solving itself, from the very beginning AI pioneers have attempted to solve the problem of automated game-playing, and later the problem of automated of reasoning and theorem proving, these problem are now viewed as problem internal to AI, not much work is presently begin done in this area.

Today the genuine AI problem originate from the quest and the efforts of scientists and engineering to develop new method for solving existing ,as well as new problem ,this is an inevitable occurrence with the perpetual technological progress we are witnessing, the associated problem to be solved, requiring the method of intelligence can be classified as problem of:

- Natural language processing
- Pattern recognition
- Computer robotics
- Expert system

1.7. Summary

Artificial intelligent system, in this title of our chapter to many people asking there self some questions are we concerned with thinking or behavior? Do we want to model humans; it's possible and many think from that way.

In this chapter we have defined and established AI background, by considering the meaning of AI history and problems

And also we had considering the solution that it was funded for that by using AI progress theoretical basis method that understanding the theoretical basis for intelligence has gone hand in hand with improvements in the capabilities of real system.

{"The scientific understanding of the mechanisms underlying thought and intelligent behavior and their embodiment in machines" by John McCarthy1956}.

Chapter TWO

EXPERT SYSTEM

2.1. Overview

So far we have talked a lot about how we can represent knowledge, but not so much about how we can use it to solve real practical problems. This chapter will therefore look at how some of the techniques discussed so far are used in expert system- systems that provide expert quality advice, diagnoses and recommendations given real world problems.

2.2. Introduction of expert system

Expert systems are meant to solve real problems, which normally would require a specialized human expert (such as a doctor or a mineralogist). Building an expert system therefore first involves extracting the relevant knowledge from the human expert. Such knowledge is often heuristic in nature, based on useful ``rules of thumb" rather than absolute certainties. Extracting it from the expert in a way that can be used by a computer is generally a difficult task, requiring its own expertise. A knowledge engineer has the job of extracting this knowledge and building the expert system knowledge base. A first attempt at building an expert system is unlikely to be very successful. This is partly because the expert generally finds it very difficult to express exactly what knowledge and rules they use to solve a problem. Much of it is almost subconscious, or appears so obvious they don't even bother mentioning it. Knowledge acquisition for expert systems is a big area of research, with a wide variety of techniques developed. However, generally it is important to develop an initial prototype based on information extracted by interviewing the expert, and then iteratively refine it based on feedback both from the expert and from potential users of the expert system, Expert systems have been used to solve a wide range of problems in domains such as medicine, mathematics, engineering, geology, computer science, business, law, defense and education. Within each domain, they have been used to solve problems of different types. Types of problem involve diagnosis e.g., of a system fault, disease or student error; design of a computer systems, hotel etc; and interpretation of, for example, geological data. The appropriate problem solving technique tends to depend more on the problem type than on the domain. Whole books have been written on how to choose your knowledge representation and reasoning methods given characteristics of your problem.

2.3. What is expert system?

One of the results of research in the area of artificial intelligence has been the development of techniques, which allow the modeling of information at higher levels of abstraction. These techniques are embodied in languages or tools, which allow programs to be built that closely, resemble human logic in their implementation and are therefore easier to develop and maintain. These programs, which emulate human expertise in well-defined problem domains, are called expert systems. The availability of expert system tools, such as CLIPS, has greatly reduced the effort and cost involved in developing an expert system. Rule-based programming is one of the most commonly used techniques for developing expert systems. In this programming paradigm, rules are used to represent heuristics, or "rules of thumb," which specify a set of actions to be performed for a given situation. A rule is composed of an if portion and a then portion. The if portion of a rule is a series of patterns which specify the facts (or data) which cause the rule to be applicable. The process of matching facts to patterns is called pattern matching. The expert system tool provides a mechanism, called the inference engine, which automatically matches facts against patterns and determines which rules are applicable. The if portion of a rule can actually be thought of as the whenever portion of a rule since pattern matching always occurs whenever changes are made to facts. The then portion of a rule is the set of actions to be executed when the rule is applicable. The actions of applicable rules are executed when the inference engine is instructed to begin execution. The inference engine selects a rule and then the actions of the selected rule are executed (which may affect the list of applicable rules by adding or removing facts). The inference engine then selects another rule and executes its actions. This process continues until no applicable rules remain

2.4. Expert system definitions

Definitions of expert systems vary. Some definitions are based on function. Some definitions are based on structure. Some definitions have both functional and structural components. Many early definitions assume rule-based reasoning

Functional Components:

what the system does (rather than how) "a computer program that behaves like a human expert in some useful ways." [Winston & Prendergast, 1984, p.6] [8]

Problem area:

"solve problems efficiently and effectively in a narrow problem area." [Waterman, 1986, p.xvii]

" Typically, pertains to problems that can be symbolically represented" [Liebowitz, 1988, p.3] [9]

Problem difficulty:

" apply expert knowledge to difficult real world problems" [Waterman, 1986, p.18]

" solve problems that are difficult enough to require significant human expertise for their solution" [Edward Feigenbaum in Harmon & King, 1985, p.5]

" address problems normally thought to require human specialists for their solution" [Michaelsen et al, 1985, p. 303]. [10]

Performance requirement:

"the ability to perform at the level of an expert" [Liebowitz, 1988, p.3]

" programs that mimic the advice-giving capabilities of human experts." [Brule, 1986, p.6]

"Matches a competent level of human expertise in a particular field." [Bishop, 1986, p.38]

"Can offer intelligent advice or make an intelligent decision about a processing function." [British Computer Society's Specialist Group in Forsyth, 1984, pp.9-10]

"Allows a user to access this expertise in a way similar to that in which he might consult a human expert, with a similar result." [Edwards and Connell, 1989, p.3] [11]

Explain reasoning

"the capability of the system, on demand, to justify its own line of reasoning in a manner directly intelligible to the enquirer." [British Computer Society's Specialist Group in Forsyth, 1984, p.9-10]

"Incorporation of explanation processes" [Liebowitz, 1988, p.3][12]

"Expert systems are computer programs mimicking the decision-making processes of humans in a limited area of expertise." (Morgan, 1997). Library applications of expert systems typically include a logical question and answer process or series of menus, a matching of user answers with appropriate information sources, a list of recommended sources, and, in some cases, a way to redirect users after mistakes

2.5. Definition Expert System Building Tools

An expert system tool, or shell, is a software development environment containing the basic components of expert systems. Associated with a shell is a prescribed method for building applications by configuring and instantiating these components. Some of the generic components of a shell are shown in Figure 2.1and described below. The core components of expert systems are the knowledge base and the reasoning engine.

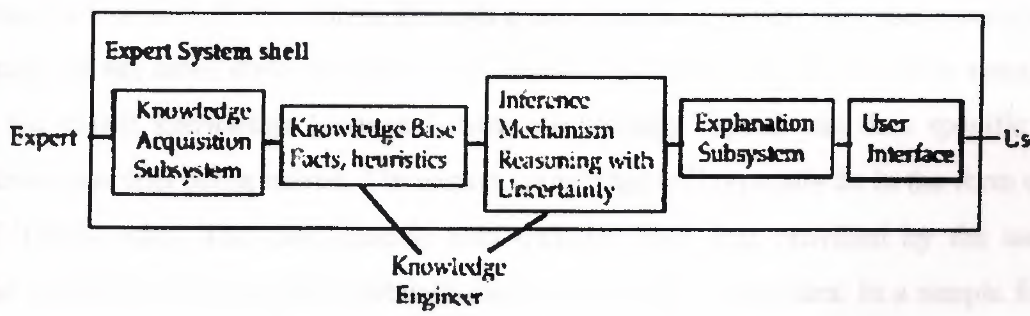


Figure 2.1. Basic Components of Expert System Tools

Knowledge base: A store of factual and heuristic knowledge. An ES tool provides one or more knowledge representation schemes for expressing knowledge about the application domain. Some tools use both frames (objects) and IF-THEN rules. In PROLOG the knowledge is represented as logical statements.

Reasoning engine: Inference mechanisms for manipulating the symbolic information and knowledge in the knowledge base to form a line of reasoning in solving a problem. The inference mechanism can range from simple modus pone's backward chaining of IF-THEN rules to case-based reasoning.

Knowledge acquisition subsystem: A subsystem to help experts build knowledge bases. Collecting knowledge needed to solve problems and build the knowledge base continues to be the biggest bottleneck in building expert systems.

Explanation subsystem: A subsystem that explains the system's actions. The explanation can range from how the final or intermediate solutions were arrived at to justifying the need for additional data.

User interface: The means of communication with the user. The user interface is generally not a part of the ES technology, and was not given much attention in the past. However, it is now widely accepted that the user interface can make a critical difference in the perceived utility of a system regardless of the system's performance.

2.6. Expert System Architecture

The user interacts with the system through a *user* interface, which may use menus, natural language or any other style of interaction). Then an inference engine is used to reason with both the expert knowledge (extracted from our friendly expert) and data specific to the particular problem being solved. The expert knowledge will typically be in the form of a set of IF-THEN rules. The case specific data includes both data provided by the user and partial conclusions (along with certainty measures) based on this data. In a simple forward chaining rule-based system the case specific data will be the elements in working memory.

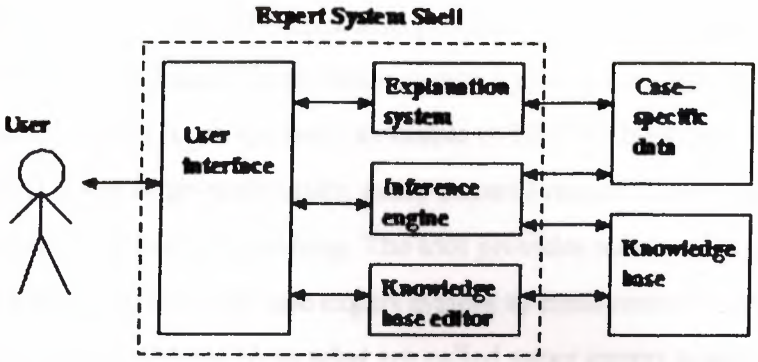


Figure 2.2 most important modules that make up

A rule-based expert system

One important feature of expert systems is the way they (usually) separate domain specific knowledge from more general-purpose reasoning and representation techniques. The general-purpose bit (in the dotted box in the figure) is referred to as an *expert system shell.* As we see in the figure, the shell will provide the inference engine (and knowledge representation scheme), a user interface, an explanation system and sometimes a knowledge base editor. Given a new kind of problem to solve (say, car design), we can usually find a shell that provides the right sort of support for that problem, so all we need to do is provide the expert knowledge. There are numerous commercial expert system shells, each one appropriate for a slightly different range of problems. (Expert systems work in industry includes both writing expert system shells and writing expert systems using shells.) Using shells to write expert systems generally greatly reduces the cost and time of development (compared with writing the expert system from scratch).

2.7. Profile of a Tool: ES/KERNEL2

ES/KERNEL2, the new version of the current best-selling tool, is geared to the development of large-scale applications It gives the application developer's choice in the use of reasoning methods: rule-based reasoning, object-oriented reasoning, and assumption-based reasoning can all be used within a single expert system. Associated with each reasoning method is a knowledge representation scheme best suited to it? For example, for object-oriented reasoning, knowledge is represented as frames, slots, and methods as it shown in Figure 2.3. ES/KERNEL2 also provides some advanced capabilities such as ATMS (Assumption-based Truth Maintenance System) and case-based reasoning (under development). Fuzzy logic has been available in ES/KERNEL and will be a part of ES/KERNEL2.For large-scale tasks; many expert systems can be connected to perform multi-layered cooperative reasoning. The tool provides a means, in the form of a blackboard data structure, for one expert system to communicate with another. The cooperating expert systems form what are called super expert systems, which in turn can cooperate with each other to solve still larger problems.

2.7.1. Seminal characteristics of ES/KERNEL2 include:

- In place of a knowledge acquisition system intended for experts' use ES/KERNEL2 provides other tools, such as a knowledge editor, to help knowledge engineers enter and modify the knowledge base.
- It provides graphic, as well as multi-media, tools for building the end-user interface. These functionalities are built with X-windows.
- The interface to external databases is designed to allow general-purpose database software to be entered into the system as frames and used in the reasoning process. Conversely, the results of the reasoning process can be stored in the database.

One objective of the ES/KERNEL environment is ease of use. For example, knowledge can be expressed in English or Japanese. If the user wants to know language specifications or grammar while editing, an explanation of a particular term and usage examples can be displayed. Reportedly, more than 50 percent of an ES developer's time is spent developing the end-user graphic interface. ES/KERNEL2 provides a variety of graphic templates and edits functions for the development of the interface.

Another objective is efficiency. A translator converts knowledge into an easy-to-process intermediate language during development, and for the production version a compiler converts the developed knowledge into a format executable at high speed. An extended RETE algorithm matches rules and objects to speed up production system inference. Other features, such as incremental compilation and knowledge partitioning, also save development time.

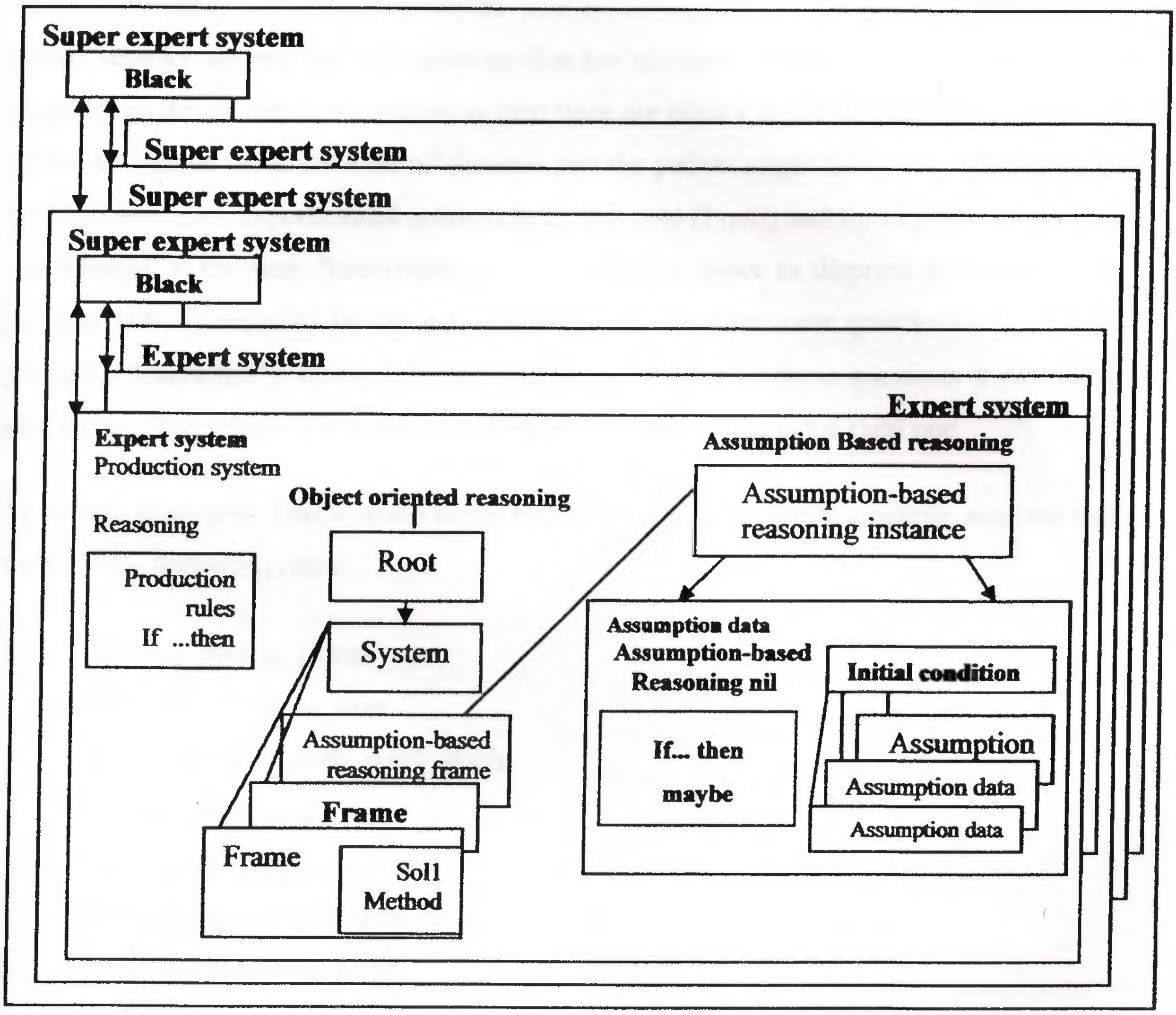


Figure 2.3. ES/KERNEL2, A Tool for Building Multi-Layered, Cooperative Expert Systems

2.8. Rules and Expert Systems

In this section we will show how expert systems based on IF-THEN rules work, and present a very simple expert system shell in Prolog, Rule-based systems can be either goal driven using backward chaining to test whether some hypothesis is true, or data driven, using forward chaining to draw new conclusions from existing data. Expert systems may use either or both strategies, but the most common is probably the goal driven/backward chaining strategy. One reason for this is that normally an expert system will have to collect information about the problem from the user by asking those questions - by using a goal driven strategy we can just ask questions that are relevant to a hypothesized solution. In a simple goal-driven rule-based expert system there are often a set of possible solutions to the problem - maybe these are a set of illnesses that the patient might have. The expert system will consider each hypothesized solution (e.g., has cold (Fred)) and try to prove whether or not it might be the case. Sometimes it won't be able to prove or disprove something from the data initially supplied by the user, so it will ask the user some questions (e.g., ``have you got a headache?"). Using any initial data plus answers to these questions it should be able to conclude which of the possible solutions to the problem is the right one

A Simple Example: This is much better explained through a simple example, suppose that we have the following rules:

1. IF engine_getting_petrol
   AND engine_turns_over
   THEN problem_with_spark_plugs
2. IF NOT engine_turns_over
   AND NOT lights_come_on
   THEN problem_with_battery
3. IF NOT engine_turns_over
   AND lights_come_on
   THEN problem_with_starter
4. IF petrol_in_fuel_tank..............THEN engine_getting_petrol

Our problem is to work out what's wrong with our car given some observable symptoms. There are three possible problems with the car: problem_with_spark_plugs, problem_with_battery, problem_with_starter. We'll assume that we have been provided with no initial facts about the observable symptoms, in the simplest goal-directed system we would try to prove each hypothesized problem (with the car) in turn. First the system would try to prove ``problem_with_spark_plugs". Rule 1 is potentially useful, so the system would set the new goals of proving ``engine_getting_petrol" and ``engine_turns_over". Trying to prove the first of these, rule 4 can be used; with new goal of proving ``petrol_in_fuel_tank" there are no rules, which conclude this (and the system doesn't already know the answer), so the system will ask the user:

Is it true that there's petrol in the fuel tank?

Let's say that the answer is yes. This answer would be recorded, so that the user doesn't get asked the same question again. Anyway, the system now has proved that the engine is getting petrol, so now wants to find out if the engine turns over. As the system doesn't yet know whether this is the case, and as there are no rules, which conclude this, the user will be asked:

Is it true that the engine turns over?

Lets say this time the answer is no. There are no other rules which can be used to prove ``problem_with_spark_plugs" so the system will conclude that this is not the solution to the problem, and will consider the next hypothesis: problem_with_battery. It is true that the engine does not turn over (the user has just said that), so all it has to prove is that the lights don't come one. It will ask the user

Is it true that the lights come on?

Suppose the answer is no. It has now proved that the problem is with the battery. Some systems might stop there, but usually there might be more than one solution, (e.g., more than one fault with the car), or it will be uncertain which of various solutions is the right one. So usually all hypotheses are considered. It will try to prove ``problem_with_starter", but given the existing data (the lights come on) the proof will fail, so the system will conclude that the problem is with the battery. A complete interaction with our very simple system might be:

System: Is it true that there's petrol in the fuel tank?

User: Yes.

System: Is it true that the engine turns over?

User: No.

System: Is it true that the lights come on?

User: No.

System: I conclude that there is a problem with battery.

Note that in general, solving problems using backward chaining involves searching through all the possible ways of proving the hypothesis, systematically checking each of them. A common way of doing this search is the same as in Prolog - depth first search with backtracking.

2.9. Advantages and Disadvantages of Expert Systems

In this section of our booklet we present some of the advantages and disadvantages of existing expert systems.

2.9.1. Advantages of Expert Systems

Permanence - Expert systems do not forget, but human experts may.

Reproducibility - Many copies of an expert system can be made, but training new Human expert is time-consuming and expensive.

Efficiency -can increase throughput and decrease personnel costs.

Although expert systems are expensive to build and maintain, they are inexpensive to operate.

Development and maintenance costs can be spread over many users.

The overall cost can be quite reasonable when compared to expensive and scarce human experts.

Cost savings:

Wages - (elimination of a room full of clerks) other costs - (minimize loan loss)

Consistency – With expert systems similar transactions handled in the same way. This system will make comparable recommendations for like situation. Humans are influenced by:

Decency effects (most recent information having disproportionate impact)

Primacy effects (early information dominates the judgment).

Documentation - An expert system can provide permanent documentation of the decision process.

Completeness - An expert system can review all the transactions, a human expert can only review a sample.

Breadth - The knowledge of multiple human experts can be combined to give a system more breadth that a single person is likely to achieve.

Reduce risk of doing business Consistency of decision-making.

Documentation.

Achieve expertise.

Entry barriers - Expert systems can help a firm create entry barriers for potential competitors.

Differentiation - In some cases, an expert system can differentiate a product or can be related to the focus of the firm (XCON).

Computer programs are best in those situations where there is a structure that is noted as previously existing or can be elicited.

2.9.2. Disadvantages of Rule-Based Expert Systems

Common sense - In addition to a great deal of technical knowledge, human experts have common sense. It is not known how to give expert systems common sense.

Creativity - Human experts can respond creatively to unusual situations, expert systems cannot

Learning - Human experts automatically adapt to changing environments; expert systems must be explicitly updated. Case-based reasoning and neural networks are methods that can incorporate learning.

Sensory Experience - Human experts have available to them a wide range of sensory experience; expert systems are currently dependent on symbolic input.

Degradation - Expert systems are not good at recognizing when no answer exists or when the problem is outside their area of expertise.

2.10. Summary

In this chapter we have give on explain of Expert systems, Expert systems are computer programs mimicking the decision-making processes of humans in a limited area of expertise. (Morgan, 1997). Library applications of expert systems typically include a logical question and answer process or series of menus, a matching of user answers with appropriate information sources, a list of recommended sources, and, in some cases, a way to redirect users after mistakes.

Most expert systems are developed via specialized software tools called shells. These shells come equipped with an inference mechanism (backward chaining, forward chaining, or both), and require knowledge to be entered according to a specified format (all of which might lead some to categorize OPS5 as a shell). They typically come with a number of other features, such as tools for writing hypertext, for constructing friendly user interfaces, for manipulating lists, strings, and objects, and for interfacing with external programs and databases.

CHAPTER THREE

FUZZY SYSTEM

3.1. Overview

Formal control logic is based in the teachings of Aristotle, where an element either is or is not a member of a particular set. Since many of the objects encountered in the real world do not fall into precisely defined membership criteria and in this chapter of our project we hade explain and find the most particular science method to performing such problem.

3.2. Introduction to fuzzy system

Many decision-making and problem-solving tasks are too complex to be understood quantitatively, however, people succeed by using knowledge that is imprecise rather than precise. Fuzzy set theory, originally introduced by Lotfi Zadeh in the 1960's, resembles human reasoning in its use of approximate information and uncertainty to generate decisions. It was specifically designed to mathematically represent uncertainty and vagueness and provide formalized tools for dealing with the imprecision intrinsic to many problems. By contrast, traditional computing demands precision down to each bit. Since knowledge can be expressed in a more natural by using fuzzy sets, many engineering and decision problems can be greatly simplified.

Fuzzy set theory implements classes or groupings of data with boundaries that are not sharply defined (i.e., fuzzy). Any methodology or theory implementing "crisp" definitions such as classical set theory, arithmetic, and programming, may be "justified" by generalizing the concept of a crisp set to a fuzzy set with blurred boundaries. The benefit of extending crisp theory and analysis methods to fuzzy techniques is the strength in solving real-world problems, which inevitably entail some degree of imprecision and noise in the variables and parameters measured and processed for the application. Accordingly, linguistic variables are a critical aspect of some fuzzy logic applications, where general terms such a "large," "medium," and "small" are each used to capture a range of numerical values. While similar to conventional quantization, fuzzy logic allows these stratified sets to overlap (e.g., a 85 kilogram man may be classified in both the "large" and "medium" categories, with varying degrees of belonging or membership to each group). Fuzzy set theory encompasses fuzzy logic, fuzzy arithmetic, fuzzy mathematical programming, fuzzy topology, fuzzy graph theory, and fuzzy data analysis, though the term fuzzy logic is often used to describe all of these.

Fuzzy logic emerged into the mainstream of information technology in the late 1980's and early 1990's. Fuzzy logic is a departure from classical Boolean logic in that it implements soft linguistic variables on a continuous range of truth values which allows intermediate values to be defined between conventional binary. It can often be considered a superset of Boolean or "crisp logic" in the way fuzzy set theory is a superset of conventional set theory. Since fuzzy logic can handle approximate information in a systematic way, it is ideal for controlling nonlinear systems and for modeling complex systems where an inexact model exists or systems where ambiguity or vagueness is common. A typical fuzzy system consists of a rule base, membership functions, and an inference procedure. Today, fuzzy logic is found in a variety of control applications including chemical process control, manufacturing, and in such consumer products as washing machines, video cameras, and automobiles. [13]

Fuzzy logic starts with and builds on a set of user-supplied human language rules. The fuzzy systems convert these rules to their mathematical equivalents. This simplifies the job of the system designer and the computer, and results in much more accurate representations of the way systems behave in the real world.

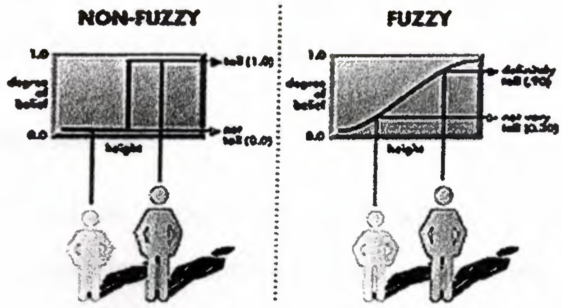


Figure 3.1. A set of user-supplied human language

3.3. What Is Fuzzy Logic?

Fuzzy logic is a powerful problem-solving methodology with a myriad of applications in embedded control and information processing. Fuzzy provides a remarkably simple way to draw definite conclusions from vague, ambiguous or imprecise information. In a sense, fuzzy logic resembles human decision making with its ability to work from approximate data and find precise solutions.

Unlike classical logic which requires a deep understanding of a system, exact equations, and precise numeric values, Fuzzy logic incorporates an alternative way of thinking, which allows modeling complex systems using a higher level of abstraction originating from our knowledge and experience. Fuzzy Logic allows expressing this knowledge with subjective concepts such as very hot, bright red, and a long time which are mapped into exact numeric ranges.

Fuzzy Logic has been gaining increasing acceptance during the past few years. There are over two thousand commercially available products using Fuzzy Logic, ranging from washing machines to high speed trains. Nearly every application can potentially realize some of the benefits of Fuzzy Logic, such as performance, simplicity, lower cost, and productivity.

Fuzzy Logic has been found to be very suitable for embedded control applications. Several manufacturers in the automotive industry are using fuzzy technology to improve quality and reduce development time. In aerospace, fuzzy enables very complex real time problems to be tackled using a simple approach. In consumer electronics, fuzzy improves time to market and helps reduce costs. In manufacturing, fuzzy is proven to be invaluable in increasing equipment efficiency and diagnosing malfunctions.

3.3.1 What is Fuzzy Logic used for?

Fuzzy logic has been used in fuzzy controllers which are widely used in control applications including refrigerators, washing machines, welding machines, cameras and robots; Fault and failure diagnosis, image processing, pattern classifying, traffic problems, collision avoidance, decision support, project planning, fraud detection and in conjunction with neural nets and expert systems.

3.3.2. Who uses Fuzzy Logic?

The Japanese use fuzzy logic controllers widely in their consumer goods. Electrical, mechanical and process engineers, equipment designers, managers, planners, data base designers, neural network users.

3.3.3. Why is Fuzzy Logic Better?

Fuzzy logic is an extension of Boolean logic into the real world where many events are more accurately described by continuous logic.

3.4 Fuzzy Sets

Fuzzy Set Theory was formalized by Professor Lofti Zadeh at the University of California in 1965. What Zadeh proposed is very much a paradigm shift that first gained acceptance in the Far East and its successful application has ensured its adoption around the world.

A paradigm is a set of rules and regulations which defines boundaries and tells us what to do to be successful in solving problems within these boundaries. For example the use of transistors instead of vacuum tubes is a paradigm shift - likewise the development of Fuzzy Set Theory from conventional bivalent set theory is a paradigm shift. Bivalent Set Theory can be somewhat limiting if we wish to describe a 'humanistic' problem mathematically. For example, Figure3.2 below illustrates bivalent sets to characterize the temperature of a room.

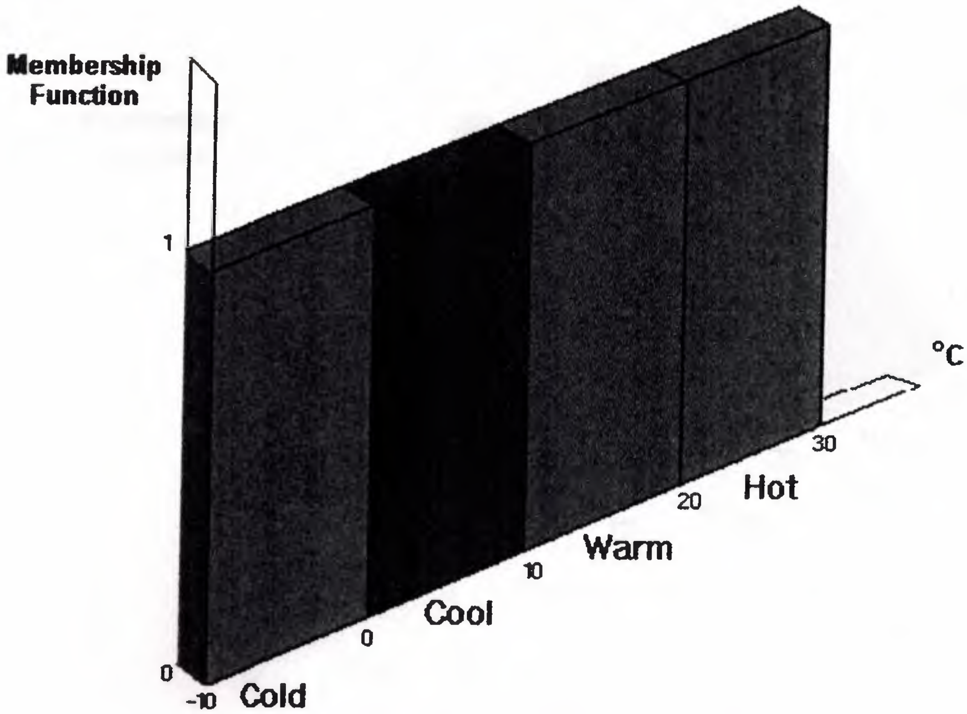


Figure 3.2. bivalent sets to character the tempt .of a room

The most obvious limiting feature of bivalent sets that can be seen clearly from the diagram is that they are mutually exclusive - it is not possible to have membership of more than one set (opinion would widely vary as to whether 50 degrees Fahrenheit is 'cold' or 'cool' hence the expert knowledge we need to define our system is mathematically at odds with the humanistic world). Clearly, it is not accurate to define a transition from a quantity such as 'warm' to 'hot' by the application of one degree Fahrenheit of heat. In the real world a smooth (unnoticeable) drift from warm to hot would occur. This natural phenomenon can be described more accurately by Fuzzy Set Theory. Figure3.3. below shows how fuzzy sets quantifying the same information can describe this natural drift.

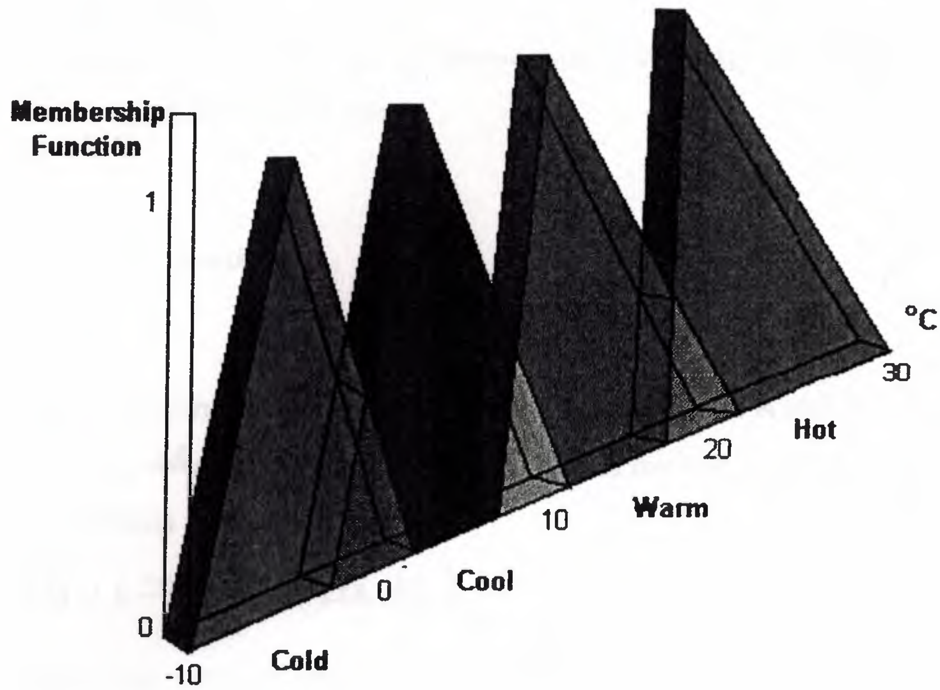


Figure 3.3. Fuzzy sets to characterize the temp of a room

3.4.1. Definitions.

- Universe of Discourse

  The Universe of Discourse is the range of all possible values for an input to a fuzzy system.

- Fuzzy Set

  A Fuzzy Set is any set that allows its members to have different grades of membership (membership function) in the interval [0, 1].

- Support

  The Support of a fuzzy set F is the crisp set of all points in the Universe of Discourse U such that the membership function of F is non-zero.

- Crossover point

  The Crossover point of a fuzzy set is the element in U at which its membership function is 0.5.

- Fuzzy Singleton

  A Fuzzy singleton is a fuzzy set whose support is a single point in U with a membership function of one

3.4.2 Fuzzy Set Operations.

- Union

  The membership function of the Union of two fuzzy sets A and B with membership functions μ_A and μ_B respectively is defined as the maximum of the two individual membership functions as in fig.3.3

$$\mu_{A \cup B} = \max(\mu_A, \mu_B)$$

The Union operation in Fuzzy set theory is the equivalent of the OR operation in Boolean algebra.

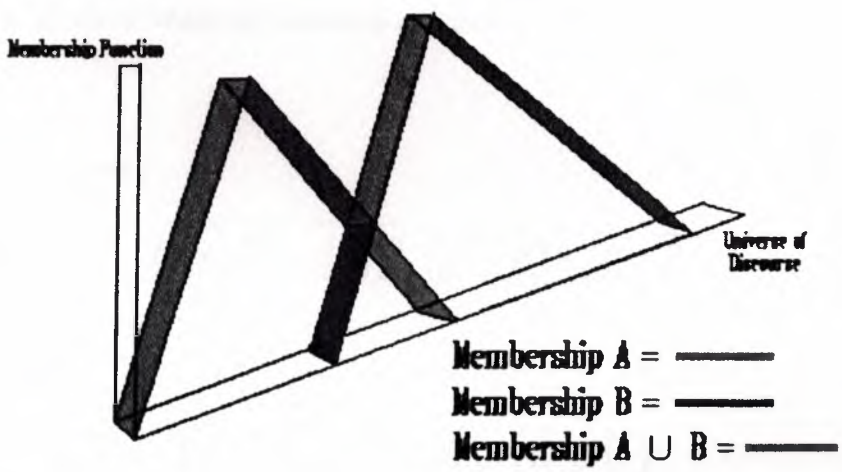


Figure 3.4. The membership function of the Union of two fuzzy sets

- **Complement**

The membership function of the Complement of a Fuzzy set A with membership function μ_A is defined as show in fig 3.4

$$\mu_{\bar{A}} = 1 - \mu_A$$

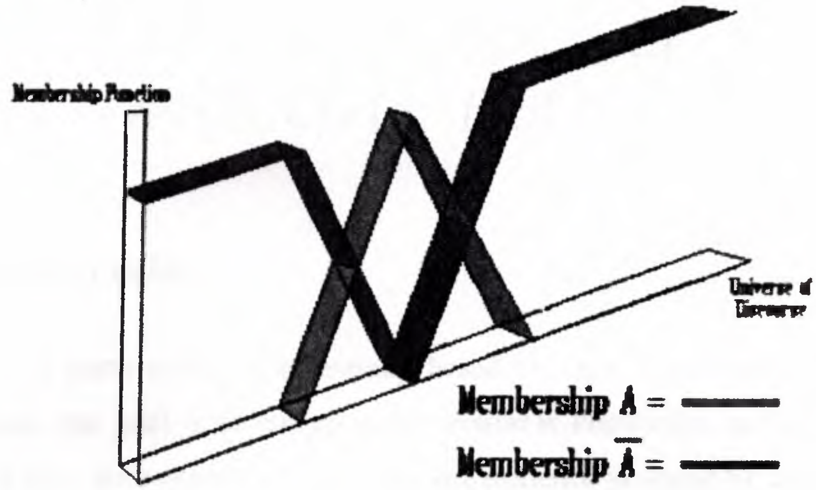


Figure 3.5. The membership function of the Complement of a Fuzzy

The following rules which are common in classical set theory also apply to Fuzzy set theory.

▪ De Morgan's law

$$\overline{(A \cap B)} = \bar{A} \cap \bar{B}, \quad \overline{(A \cup B)} = \bar{A} \cap \bar{B}$$

▪ Associatively

$$(A \cup B) \cup C = A \cup (B \cup C)$$
$$(A \cap B) \cap C = A \cap (B \cap C)$$

• Commutatively

$$A \cap B = B \cap A, \quad A \cup B = B \cup A$$

- Distributives

$$A\cup(B\cap C)=(A\cup B)\cap(A\cup C)$$
$$A\cap(B\cup C)=(A\cap B)\cup(A\cap C)$$

3.5. Generating fuzzy rules

The construction of a fuzzy model is essentially based on data and/or expertise on the system. In both cases our goal is to compress the available knowledge and/or data in a manner that enables us to state general assertions on the evidence provided by simple, small and comprehensible knowledge bases.

If empiric data is available, we should aim to using a sample which well represents the re-sportive population. The sample size depends on the rule generation algorithm, the size of the population, the homogeneity of the data, the dimensionality of the model space, the objective of the model construction and the established error limits of the system, inter alias.

For example, when generating the rules, the *grid partition technique*, which generates rules by using all the combinations of input and output values, usually requires large sample sizes and heavy computation. If a one-input-one-output system requires at least ten data points, a four-variable system should use 10^3=1000 data points. It also would yield large rule bases if several variables are included in the system. Four variables, each using three values, already yield 34=81 combinations (i.e., system) as it in figure (3.5, 6)

If the sample size is sufficiently large, we may overcome several of the preceding problems by dividing the data into two parts, training data, and control data. The model construction and its possible tuning are based on the former set, whereas we assess this model by testing it with the latter set (and vice versa, if necessary). Then, our model can yield good outputs on a general level. A typical over determination problem occurs when the model fits the training data well, but clearly yields unsatisfactory outputs for the control data. We must also bear in mind that these models are only appropriate to interpolation, whereas extrapolation is problematic to any model.

If appropriate data is unavailable and we only utilize expertise, the system construction may be more problematic from the standpoint of generalization and applicability. We can also utilize expertise to support the data or vice versa. Below the standard normal distribution is depicted. By using the scatter technique (SubclusterC algorithm), we may generate fuzzy cluster centers (rules), and the number the rules is dependent on our goal and problem-setting.

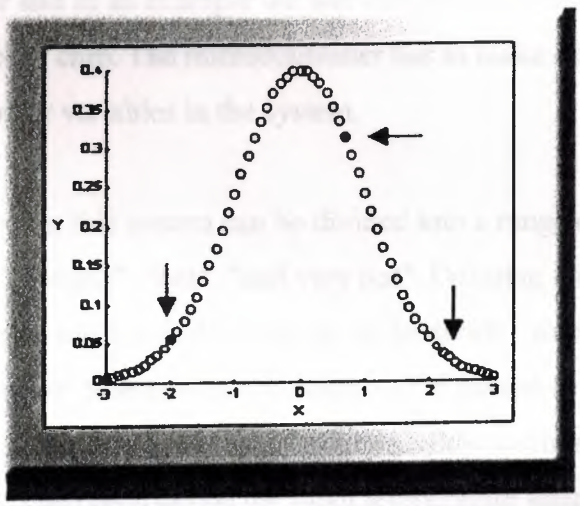


Figure 3.6. Three Fuzzy Cluster Centers (Rules)

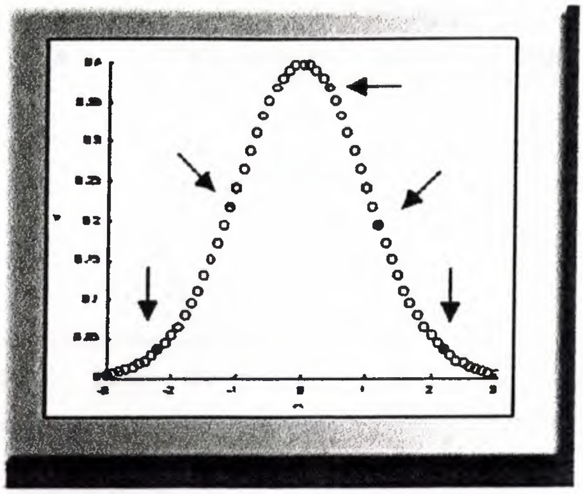


Figure 3.7. Five Fuzzy Cluster Centers (Rules)

3.6. Example of fuzzy modeling

In this part of our chapter and as an example we will consider an antilock braking system, directed by a microcontroller chip. The microcontroller has to make decisions based on brake temperature, speed, and other variables in the system.

The variable "temperature" in this system can be divided into a range of "states", such as: "cold", "cool", "moderate", "warm", "hot", "and very hot". Defining the bounds of these states is a bit tricky. An arbitrary threshold might be set to divide "warm" from "hot", but this would result in a discontinuous change when the input value passed over that threshold. The way around this is to make the states "fuzzy", that is, allow them to change gradually from one state to the next. You could define the input temperature states using "membership functions" such as the following:

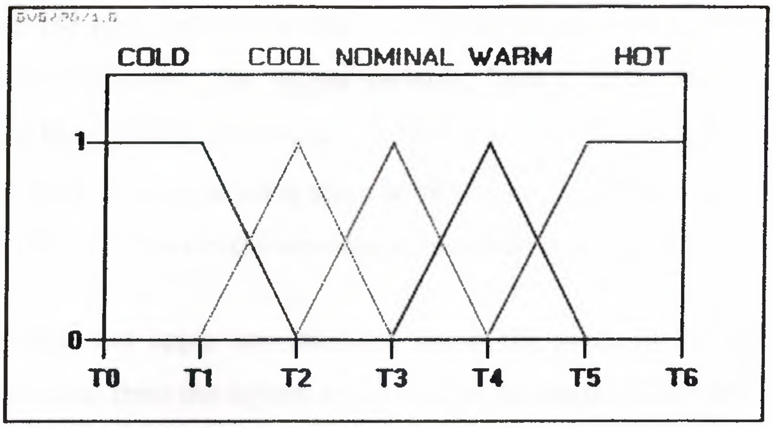


Figure 3.8. Variable "temperature divided into a range of "states

With this scheme, the input variable's state no longer jumps abruptly from one state to the next. Instead, as the temperature changes, it loses value in one membership function while gaining value in the next. At any one time, the "truth value" of the brake temperature will almost always be in some degree part of two membership functions: 0.6 nominal and 0.4 warm, or 0.7 nominal and 0.3 cool, and so on.

The input variables in a fuzzy control system are in general mapped into by sets of membership functions similar to this, known as "fuzzy sets". The process of converting a crisp input value to a fuzzy value is called "fuzzification".

A control system may also have various types of switch, or "ON-OFF", inputs along with its analog inputs, and such switch inputs of course will always have a truth value equal to either 1 or 0, but the scheme can deal with them as simplified fuzzy functions that are either one value or another.

Given "mappings" of input variables into membership functions and truth values, the microcontroller then makes decisions for what action to take based on a set of "rules", each of the form:

```
IF brake temperature IS warm AND speed IS not very fast
THEN brake pressure IS slightly decreased.
```

In this example, the two input variables are "brake temperature" and "speed" that have values defined as fuzzy sets. The output variable, "brake pressure", is also defined by a fuzzy set that can have values like "static", "slightly increased", "slightly decreased", and so on. This rule by itself is very puzzling since it looks like it could be used without bothering with fuzzy logic, but remembers the decision is based on a *set* of rules:

- *All* the rules that apply are invoked, using the membership functions and truth values obtained from the inputs, to determine the result of the rule.
- This result in turn will be mapped into a membership function and truth value controlling the output variable.
- These results are combined to give a specific ("crisp") answer, the actual brake pressure, a procedure known as "defuzzification".

This combination of fuzzy operations and rule-based "inference" describes a "fuzzy expert system", Traditional control systems are based on mathematical models in which the control system is described using one or more differential equations that define the system response to its inputs. Such systems are often implemented as "proportional-integral-derivative (PID)" controllers. They are the products of decades of development and theoretical analysis, and are highly effective.

If PID and other traditional control systems are so well-developed, why bother with fuzzy control? It has some advantages. In many cases, the mathematical model of the control process may not exist, or may be too "expensive" in terms of computer processing power and memory, and a system based on empirical rules may be more effective.

Furthermore, fuzzy logic is well suited to low-cost implementations based on cheap sensors, low-resolution analog-to-digital converters, and 4-bit or 8-bit one-chip microcontroller chips. Such systems can be easily upgraded by adding new rules to improve performance or add new features. In many cases, fuzzy control can be used to improve existing traditional controller systems by adding an extra layer of intelligence to the current control method.

3.7. The fuzzy logic control problem

A typical embedded control problem is usually solved with a controller that follows the concept of Figure 3.8 Naturally, depending on the embedded engineer's experience and budget; this underlying structure may look quite different from one design to another. In all cases, the task of the controller is to modify the Process Inputs so that the Process Outputs converge towards the Set Points. This is usually referred to as the control law, and can be implemented by several control algorithms like PID, Optimal Control, Adaptive Control, FL Control, Neural Network Control, etc.

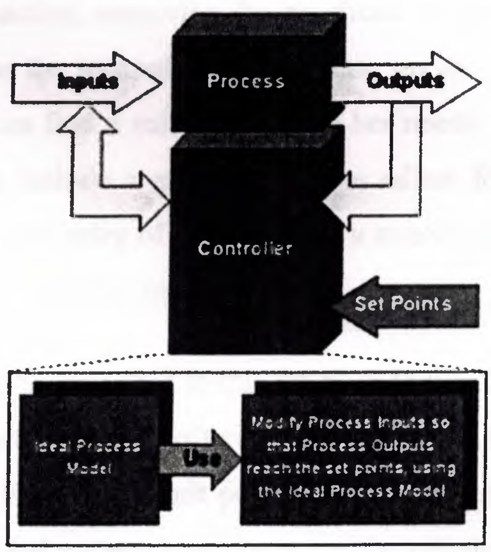


Figure 3.9. The typical controller action

Each control algorithm has advantages and disadvantages, relative to the nature of the process under control; however FL control algorithms can offer significant advantages to the embedded systems engineer, for any system:

- Straight-forward definition of the system variables, using linguistic & graphical methods,
- Rapid modeling of the process model using simple language if...then... statements,
- Easy access to the FL definition of the controller, resulting in easy debugging and tuning,
- Ability to model non-linear processes with the same ease of modeling linear ones,
- Availability of powerful FL development tools with build-in:
    - Source code generators
    - Reporting tools
    - Debugging and tuning tools.

3.8. The fuzzy logic controller development

For the development of a FL controller, it is strongly recommended to use one of the commercially available development tools. These tools address the problem to its appropriate level of abstraction, removing the overhead for one to become a fuzzy logic expert, before she can use the technology. Depending on the embedded system engineer and FL knowledge, one can find a suitable tool for her needs. 's preferences Most of the advanced tools nowadays include a graphical design editor for the definition of the FL variables, a text editor for the entry of the FL rules, a source code generator for exporting the FL design and some type of debugging method.

The battery charge controller described in this article was developed with the Fuzzy Logic Development Environment (FLDE), offered by Syndics Ltd. FLDE is one of the commercial results of the European Esprit project OMI-FEM and is designed to serve best the requirements of embedded system development in the automotive domain. The program runs under MS-Windows and generates self-contained, strict ANSI-C code, suitable for immediate compilation and embedding in an automotive system. The FL development with FLDE is quite straight-forward and is based on a step-by-step definition of three basic entities: the FL project, the FL nouns and the FL rules. The project is the reference point for all the files and entities that constitute a FL system. The nouns are the FL variables, while the rules are the textual sentences that interconnect them.

3.9. Summary

As we defined in the suppliant of this chapter that was not long ago considered another esoteric, scientific branch of the academic world, is now an all pervasive technology with hundreds of applications in control systems and decision support systems.

We have considered in this chapter the logic of how fuzzy is working by considering a small example about it as you have read, and we have considered also important method to make it easy to understand fuzzy logic.

And we have entered to fuzzy modeling by consider the way we have to use in modeling and the steps it have to be started in our modeling ,and also we consider same problem fuzzy logic control and concept action to solved some like of this kinds. And in end it become, the way of how we can develop fuzzy logic

NEAR EAST UNIVERSITY
LIBRARY
1988 - LEFKOŞA

NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

ARTIFICIAL INTELLIGENCE SYSTEM

Graduation Project

COM – 400

Student: Nader Ibraik

Supervisor: Assoc. Prof. Dr Adnan Khashman

Nicosia - 2003

ACKNOWLEDGEMENTS

First of all I am thanking full to the most gracious "ALLAH" the almighty, who enable me to complete this project

Secondly, I would like to award my supervisor Assoc. Prof. Dr Adnan Khashman for being so operative averring supervise me in this work, and for his overwhelming and limitless help he had done to me.

Thirdly, I will never over look the encourage I had resaved from all my family, specially my parents, my "father" the best father in this world, and the best sweetest woman in the word "mother", for there supporting me and caring for me, I am thankful for them and my family members from brother and sisters.

Fourthly this will be the most important moment in my life that I will settlement my project to the best brother" ZAKY-IBRAIK" because of his encourage me and supporting me in the long of my 4th year study ,I am so thankful for him, my brother I am own you too match.

Finally I like to thank the best friends I had met them in Cyprus Mohammed Asfour and Mohammad Shuqair and Mohammad Qunj who I can't express my feeling in such word that for there suggestion and evolution through out completion my project.

I will never forget them

ABSTRACT

Since the beginning, humankind has sought to use elements in the surrounding environment to make life easier and the tasks at hand more efficient. In keeping with this tradition, people have toyed with and explored the concept of using machines to solve problems since ancient times, Only in this 20th century have significant advances occurred, making the possibility of an actual manifestation of artificial intelligence more and more a reality. The following timeline provides a look at important occurrences in the development of the field of artificial intelligence. Those items in bold print are what we considered the most significant events in the development of AI.

This project presents a study of A.I. system and provides a comparison between them , Neural Network ,Fuzzy system and Expert system are investigated, and a real-life application on A.I. is presented.

TABLE OF CONTEN

ACKNOWLEDGMENT i

ABSTRACT ii

TABLE OF CONTENTS iii

INTRODUCTION 1

CHAPTER ONE: ARTIFICIAL INTELLIGENCE 3

1.1 OVERVIEW 3

1.2. WHAT IS ARTIFICIAL INTELLIGENCE SYSTEM? 3

1.2.1. A GENERAL DEFINITION 4

1.2.2. ANOTHER DEFINITION 4

1.2.3. AN ENGINEERING DEFINITION 4

1.3. ARTIFICIAL INTELLIGENCE HISTORY AND EVOLUTION 6

1.4. THE CHARACTERISTICS OF ARTIFICIAL INTELLIGENCE SYSTEM 10

1.5. ARTIFICIAL INTELLIGENCE METHOD 10

1.5.1. KNOWLEDGE REPRESENTATION 11

1.5.2. SOLUTION SEARCH METHOD 12

1.5.3. METHOD OF REASONING 12

1.5.4. MACHINE LEARNING 13

1.6. ARTIFICIAL INTELLIGENCE PROBLEMS 14

1.7 SUMMARY 14

CHAPTER TWO: EXPERT SYSTEM 16

2.1 OVERVIEW 16

2.2. INTRODUCTION OF EXPERT SYSTEM 16

2.3. WHAT IS EXPERT SYSTEM? 17

2.4. EXPERT SYSTEM DIFINITIONS 18

2.5. DIFINITIONS EXPERT SYSTEM BUILDING TOOLS 19

2.6. EXPERT SYSTEM ARCHITECTURE 20

2.7. PROFILE OF ATOOL: ES/KERNEL2 21

2.7.1. SEMINAL CHARACTERISTICS OF ES/KERNEL2 INCLUDE 22

2.8. RULES AND EXPERT SYSTEM 23

2.9. ADVANTAGES AND DISADVANTAGES OF EXPERT SYSTEM 26

2.9.1. ADVANTAGES OF EXPERT SYSTEMS 26

2.9.2. DISADVANTAGES OF EXPERT SYSTEM 28

2.10. SUMMARY 30

CHAPTER THREE: FUZZY SYSTEM 30
3.1 OVERVIEW 30
3.2. INTRODUCTION TO FUZZY SYSTEM 32
3.3. WHAT IS FUZZY LOGIC? 33
3.3.1 WHAT IS FUZZY LOGIC USED FOR? 33
3.3.2. WHO USES FUZZY LOGIC? 33
3.3.3. WHY IS FUZZY LOGIC BETTER? 33
3.4 FUZZY SETS 35
3.4.1. DEFINITIONS 36
3.4.2 Fuzzy Set Operations 38
3.5. GENERATING FUZZY RULES 40
3.6. EXAMPLE OF FUZZY MODELING 43
3.7. THE FUZZY LOGIC CONTROL PROBLEM 44
3.8. THE FUZZY LOGIC CONTROL DEVELOPMENT 43
3.9 SUMMARY 45
CHAPTER FOUR: NEURAL NETWORKS 46
4.1 OVERVIEW 46
4.2. INTRODUCTION 46
4.3. WHAT IS A NEURAL NETWORKS? 47
4.3.1 SOME DEFINITIONS OF A NEURAL NETWORK 47
4.3.2. WHO IS CONCERNED WITH NEURAL NETWORKS? 48
4.4. HISTORICAL BACKGROUND OF NEURAL NETWORK 48
4.5. HUMAN AND ARTIFICIAL NUEROL 51
4.5.1 HOE THE HUMAN BRAIN LEARNS 51
4.5.2. FROM HUMAN NEURONS TO ARTIFICIAL NEURONS 52
4.6. WHY IT IS DIFFICULT TO MODEL A BRAIN-LIKE NEURAN ETWORK 52
4.7. AN ENGINEERING APPROACH 53
4.7.1 A SIMPLE NEURON 53
4.7.2. FIRING RULES 54
4.7.3. PATTERN RECOGNITION 55
4.7.4. A MORE COMLICATED NEURON 58

4.8. ARCHITECTURE OF NEURAL NETWORK 58

4.8.1. FEED-FORWORD NETWORK 58

4.8.2 FEEDBACK NETWORK 59

4.8.3 NETWORK LAYERS 60

4.9. WHY USE NEURAL NETWORK? 61

4.10. SUMMARY 61

CHAPTER FIVE: A.I .APPLICATIO 63

5.1 OVERVIEW 63

5.2 INTRODUCTION 63

5.3 ARTIFICIAL NEUROL NETWORK 64

5.4 PROBLEM FORMULATION 66

5.5 PROPOSED METHODOLOGY 66

5.5.1 INFORMATION ANALYSIS 68

5.5.2. Model identification 69

5.6. APPLICATIPON TO SAN JUANRIVER BASIN 72

5.7. SUMMARY 76

CONCLUSION 78

REFERENCES 80

INTRODUCTION

The increasing prominence of computers has led to a new way of looking at the world. This view sees nature as a form of computation. By applying Artificial Intelligent (AI) to the computers to solve difficult problems, whose solutions require human intelligence, Together with the neural networks another interesting algorithms approach, inspired by the biological behavior; Artificial Intelligent is being applied in complicated computer systems, along with the neural networks.

The aim of this project is to show and define the Artificial Intelligent system in order to use it wisely and have a look on its problems and their solutions.

In chapter one brief history and background of Artificial Intelligent, Then a discussion of the structure of the Artificial Intelligent

In chapter two brief history of expert system along with biological terminology and their benefits and their structures. A brief discussion of their structures, topologies and their applications.

In chapter three concise fuzzy system. What is the use of fuzzy logic, with some brief discussion of their structures by using soma of fuzzy methods we will see in example?

In chapter four brief history of neural networks along with biological terminology and their benefits and their structures. A brief discussion of their structures, topologies and their applications.

In chapter five the presentation of the applications in the artificial intelligence. Dealing with examples related to applying the artificial intelligence, monthly stream flow prediction using artificial neural networks (ANN) on mountain watersheds. The

procedure addresses the Selection of input variables, the definition of model architecture and the strategy of the learning process.

CHAPTER ONE

Artificial Intelligence system

1.1. Overview

This chapter is an introduction to Artificial Intelligence, Artificial intelligence or AI shortly is a relative young discipline compared to the by now mature nature sciences such as physical or chemistry or biology and mathematics. Detailed historical background is provided. A method to choose responses according to its objectives and memories, and some way of performing these responses in and on its environment.

1.2. What is Artificial Intelligence System?

Of fundamental concern in the application of artificial intelligence is the question "what is artificial intelligence?" and providing a straightforward.

The term Artificial is perhaps simple enough to understand, this meaning contrived, synthetic, man-made, but what is intelligence? We don't really know intelligence means; the field of artificial intelligence has been in existence for approximately 40 years and provides us with a working and tracking it.

The majority of attempts to precisely define the many-faceted term artificial intelligence do not completely succeed the failure here is due to the

- Non-existence of a precise and comprehensive definition (natural) intelligence itself
- Scope and depth of artificial intelligence in terms of its wide application area and the extent of problem to be solved

  And some sources define intelligence as the
- Ability to acquire, analyze, understand and creatively apply the knowledge

- Ability to reason (think) and intelligently handle (behave)

1.2.1. A General Definition

Widely accepted definition of artificial intelligence are both controversial and elusive considering the difficulty in defining natural intelligence (NT), it is probable be better then to attempt definition of AI then the origin of AI may be traced back to conference at Dartmouth college in the summer of 1956 and the broadest definition is that:

AI is field of study that seeks to explain and emulate intelligent behavior in term of computational processes

1.2.2. Another Definition

- Artificial intelligence is activity carried out by machine that, if carried by human, would be considered intelligent. From practical point of view, simulating intelligence is just a good as actual intelligence
- Artificial intelligence is branch of computer science dealing with computer system implementing a restricted but definite part of human intelligence, particularly in knowledge acquisition, perception learning, reasoning, language and scan understanding.
- AI is the field of study that seeks to explain and emulate intelligent behavior in the terms of computational processes.

1.2.3. An Engineering Definition

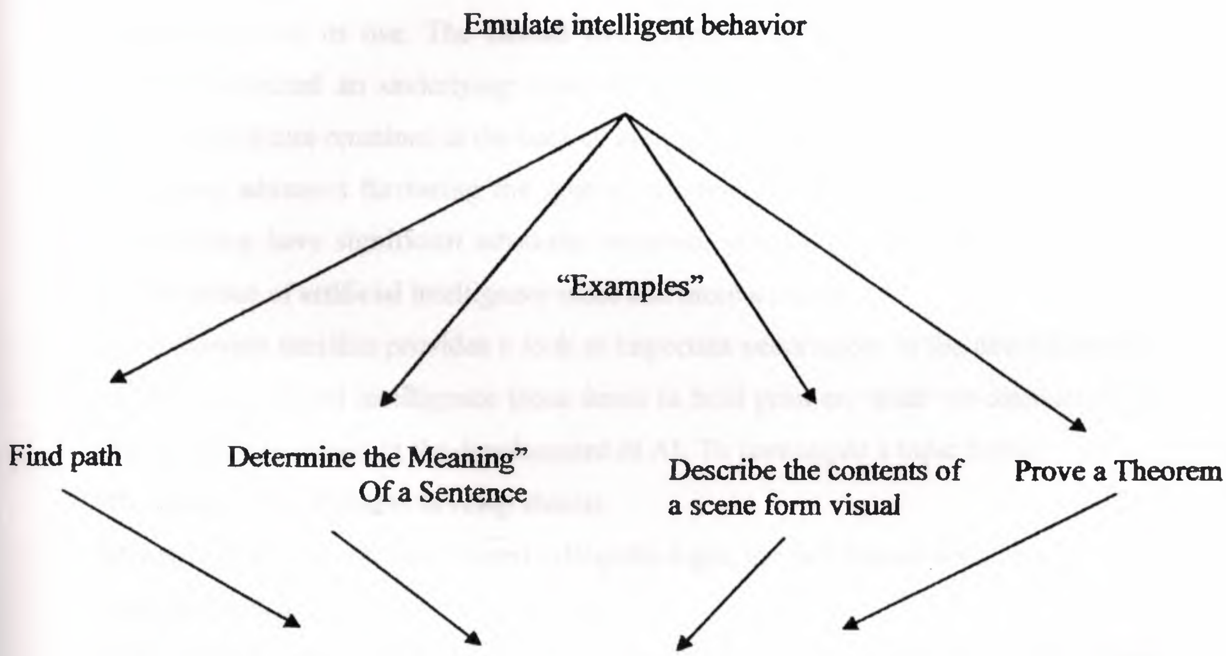
Since the applied artificial intelligence, concerned with implementation of intelligence system, is type of engineering, or at least an applied science, the question is that whether artificial intelligence itself is consequently a branch of engineering or an application field of cognitive science.

From an engineering viewpoint that artificial intelligence is about:

"Generating representation and procedures that automatically solve problem heretofore solved by humans"

An engineering approach to AI requires the development of programs; that is algorithms and databases that exhibit intelligent behavior. Figure 1.1 shows characterization of intelligent behavior without defining intelligence. Since this autonomous capability is a form of advanced computation, an alternative descriptor might be machine intelligence this reinforces our previous definition since:

- Mechanization of intelligence implies the need for an explicit and quantitative description
- Codifying expert knowledge is articulating intelligence.



These are the ramification of intelligence behavior that do not require general definition or characterization of intelligence.

Figure 1.1 AI goals via example of intelligence behavior

1.3. Artificial Intelligence History and Evolution

Since the beginning, humankind has sought to use elements in the surrounding environment to make life easier and the tasks at hand more efficient. In keeping with this tradition, people have toyed with and explored the concept of using machines to solve problems since ancient times. The development of these ideas can be seen as far back as ancient Greece, with the mention of intelligent machines in mythology (e.g., Ephesus and Pygmalion). Most people are aware of the development of calculators ("the brains of AI") throughout history. The earliest type was the abacus, which was used in China. The Egyptians invented a counting machine that used pebbles some time before Herodotus noted its use. The Greeks and Romans had similar devices. These early attempts reflected an underlying desire to replicate human reasoning in nonhuman forms. This desire remained at the back of human consciousness over the centuries, with occasional advances furthering the goal of creating 'thinking machines.' Only in this 20th century have significant advances occurred, making the possibility of an actual manifestation of artificial intelligence more and more a reality.[1]

The following timeline provides a look at important occurrences in the development of the field of artificial intelligence those items in bold print are what we considered the most significant events in the development of AI. To investigate a topic further

6th century B.C. Chinese develop abacus

5th century B.C. Aristotle invented syllogistic logic, the first formal deductive reasoning system.

13th century Talking heads were said to have been created, Roger Bacon and Albert the Great reputedly among the owners. Ramon Lull, Spanish theologian, invented machines for discovering nonmathematical truths through combinatory

15th century Invention of printing using moveable type. Gutenberg Bible printed (1456).

15th-16th century Clocks, the first modern measuring machines, were first produced using lathes.

16th century Clockmakers extended their craft to creating mechanical animals and other novelties. Rabbi Loews of Prague is said to have invented the Golem, a clay man brought to life (1580).

17th century early in the century, Descartes proposed that bodies of animals are nothing more than complex machines. Many other 17th century thinkers offered variations and elaborations of Cartesian mechanism Wilhelm Schickard (1592-1635), invented an automatic digital calculator (1633) Hobbes published The Leviathan, containing a material and combinatorial theory of thinking. Pascal created the first mechanical digital calculating machine (1642). Leibniz improved Pascal's machine to do multiplication & division (1673) and envisioned a universal calculus of reasoning by which argument could be decided mechanically

18th century the 18th century saw a profusion of mechanical toys, including the celebrated mechanical duck of Vaucanson and von Kemp Len's phony mechanical chessplayer,TheTurk(1769)

19th century Ladies (led by Ned Ladd) destroyed machinery in England (1811- 1816). Mary Shelley published the story of Frankenstein's monster (1818). George Boole developed a binary algebra representing (some) "laws of thought."

20th century Bertrand Russell and Alfred North Whitehead published Principia Mathematic, which revolutionized formal logic. Russell, Ludwig Wittgenstein, and Rudolf Carnal lead philosophy into logical analysis of knowledge, as it shown in Figure 1.2 timeline provides a look at important occurrences in the development of the field of artificial intelligence.[2]

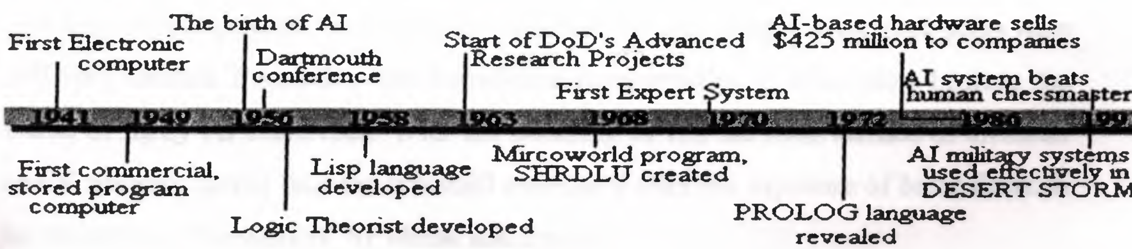


Figure 1.2 AI timeline provide

The Birth of AI (1945-56):

It was the postwar period (1945-1956) that Artificial Intelligence first emerged as a widely discussed field. What propelled the birth of Artificial Intelligence was the arrival of modern computer technology. The development of the modern computer technology affected the AI research tremendously. Many pioneers of AI broke away from the traditional approach of artificial neurons and decided that the human thought could be more efficiently emulated with modern digital computer. Those who did not accept digital computers as the new approach stayed in the parallel field of neural network [3]

The Dawning Age of AI (1956-63)

1956-1963 represents the dawning of an intensive AI wave. During this period, major AI research centers concentrated their work on two main themes. First, the attempt to limit the breadth of searches in trial-and-error problems led to the initiation of projects such as Logic Theorist (considered as the first AI program), Geometry Theorem Proved, and SAINT. Next, the study on computer learning includes projects on chess, checkers, and pattern recognition programs. Specialized list-processing AI languages such as LISP were also developed in MIT and other places in 1958.[4]

The Maturation of AI (1963-70)

By mid 60's, AI had become the common goal of thousands of different studies. AI researchers utilized their programming techniques and the improved computers in pursuing various projects. However, the memories of computers during these years were still very limited. Perception and knowledge representation in computers became the theme of many AI researches. With the booming of AI, the rival science of artificial neural network would face the downfall especially after the exposure of basic flaws in its researching "Perception" by Minsk and Popert.

The Specialization of Various AI Studies (1970's)

Different AI-related studies had developed into recognizable specialties during the 70's. Edward Feigenbaum pioneered the research on expert systems; Roger Shank promoted language analysis with a new way of interpreting the meaning of words; Marvin Minks propelled the field of knowledge representation a step further with his new structures for

representing mental constructs; Douglas Lenat explored automatic learning and the nature of heuristics; David Marr improved computer vision; the authors of PROLOG language presented a convenient higher language for AI researches. The specialization of AI in the 70's greatly strengthened the backbone of AI theories. However, AI application were still few and premature.[5]

The Unfulfilled Expectations (1980's)

The 1980's were a period of roller coasting for AI. The anti-science tradition of the public was improved greatly following the appearance of Star Wars movies and the new popularity of the personal computers. XCON, the first expert system employed in industrial world, symbolized the budding of real AI application. Within four years, XCON had grown tenfold with an investment of fifty person-years in the program and an achievement of saving about forty million dollar's in testing and manufacturing costs for the industrial clients. Following the brilliant success was the AI boom. The number of AI groups increased tremendously and in 1985, 150 companies spent about $1 billion altogether on internal AI groups. However, the fundamental AI algorithm was still unsatisfying. As Marvin Minsk warned the over-confident public: these seemingly intelligent programs simply make dumb decisions faster. Indeed, the warning foreshadowed the downfall of AI industry in late 80's. The replacing of LISP machines by standard microcomputers with AI software's in the popular C language in 1987 and the instability of expert systems caused a painful transition on expert system industry; the computer vision industry also suffered from a big setback when Machine Vision International crashed in 1988; one other major loss was the failure in Autonomous Land Vehicle project (AI drivers + Robotics). The AI industry started recovering at the end of the 80's but learning from the past experience, public assumed a much more conservative view on AI ever since. Another notable event is the revisiting of neural network with the work done by the Parallel Distributed Processing Study Group. In 1989, about three hundred companies were founded to compete for the predicted $1 billion market for neural nets by end of the century. [6]

AI Being Incorporated in War (early 1990's)

The Persian Gulf War in the early 90's proved the importance of AI research for military use. Tasks as simple as packing a transport plane and as complicated as the timing and coordination of Operation Desert Storm were assisted by AI-oriented expert systems. Advanced weapons such as "cruise missiles" were equipped with technologies previously studies in different AI-related fields such as Robotics and Machine Vision. Two projects succession the Automated Land Vehicle project were the Pilot's Associate project (electronic copilot) and the Battle Management System project (military expert systems).[7]

New AI Applications (late 1990's)

The victory of Deep Blue over chess champion Kasparov in 1996 led to a new summit of AI gaming. A new branch of expert systems has been expected to prosper as Genetic Engineering matures. Manipulating such gigantic knowledge base of human DNA map (Bioinformatics) will require very specialized algorithms and AI researches.[8]

1.4. The Characteristics of Artificial Intelligence System

- Imitation of the human reasoning process.
- Sequential information processing.
- Explicit knowledge representation.
- Use of deductive reasoning.
- Learning is outside system.

1.5. Artificial Intelligence Method

Artificial intelligence problems span abroad array of application areas of human active. The problem to be solved are not only numerous but also divers, and quit dissimilar, they require fundamentally different approaches for problem solution. But the question that consequently arises is whether there are same common, general methods of problem solving, compatible with the large spectrum of solution? And of course the answer is a great extent affirmative, the most suitable AI methods will be

briefly reviewed, in terms of the role they play in the process of problem of solving, the method are grouped into:

- Knowledge representation
- Solution search
- Reasoning
- Machine learning

1.5.1. Knowledge representation

Knowledge representations reasoning with the knowledge are two major building blocks of contemporary AI system, capturing for later essential features of a knowledge-domain in a form convenient for later knowledge processing is the first constructive step towards the building of an intelligent, Knowledge-based system belonging to the knowledge acquisition phase of the building process. Here a form has to be found for the abstract representation of facts and the relationship between the facts that will cover as much of domain knowledge as possible. In addition to confined generality, knowledge representation method should include the representation of qualitative and semantic knowledge as well as meat- knowledge with reference to this possible knowledge levels to dealt in the AI as depicted in fig1.3 should be kept in mind.

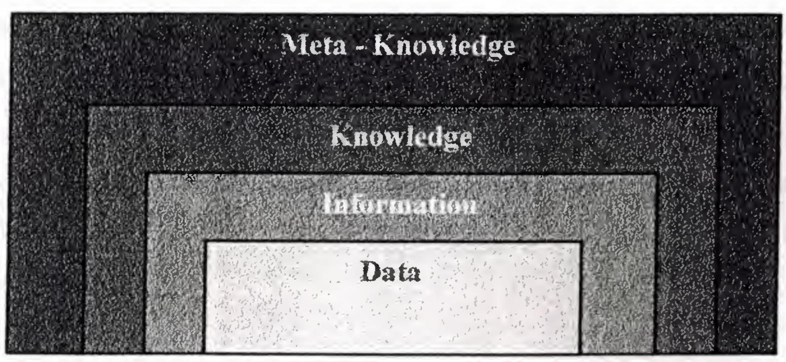


Figure 1.3 Knowledge Levels

1.5.2. Solution search method

The pivotal issues of problem solving strategies are classified to be aimed at goal finding, but before a problem can be solved it has to be exactly defined first, consequently the intrinsic problem solving method includes a:

- Problem representation
- Solution search

Method "problem representation is prerequisite for a solution search to be applied, prior to the application we must know where the total problem is allocated where we are at the beginning of the problem solution where the goal to be pursued is within the whole constellation and how to reach it" In AI a widely accepted idea that came out of the early research is that the most adequate problem formulation is its state-space representation in term of

- Initial state, from which the solutions search start.
- Terminal state i.e. the state that represents the problem solution or its goal.
- Operation i.e. starts transformation to be employed for stepwise move from the initial to the goal start.

1.5.3. Method of Reasoning

The next major group of AI is centered on the problem of Reasoning with the stored knowledge, Reasoning is actually drawing conclusion from the facts or actually inferring conclusions from premises, the method however aside from the way the knowledge engineer or the domain expert uses it for his purposes can be reduced mainly to the representation forms understanding by computer of which the following three are basic, namely

- Logic expressions
- Production rules
- Slot-and-filler structures

The most natural approach in developing reasoning method would be to first study that what is known as common sense reasoning based on common sense knowledge this is what each person starting any age can do and does: he reasons about the space and abject of his surrounding, about their shape, colors, and dimension, about the time, event and the sequence of event etc, there is same serious critical option that we may not be able build an intelligent program that will be superior to a 3 year old child ,the state-of-the –art here is that AI although able to solve domain expert problem.

On other hand, the method of reasoning of called automated reasoning has been much more successful and is used in expert system it is based on logic programming in which reasoning is already built in defined in terms of mathematical logic, automated is reasoning is a process of using some unambiguous notations for representing knowledge in order to draw corresponding inference. In diagnostic expert system there are two levels of knowledge representation of the system to be identified:

- "Shallow" knowledge representation, description the system under diagnosis by set of heuristic chunks of established facts.
- "Deep" knowledge representation that includes the description of the structure and the function behavior of the system under diagnosis.

1.5.4. Machine learning

An exact definition of learning is difficult to find, it might be any change to an intelligent system, such as the addition of any single facts, implanting of a new piece of knowledge or a control strategy, Simon (1983) says that the system changes in the process of learning directly contribute to the improvement of its efficiency in the sense of its better behavior when solving more of the same or similar problem, this imposition resembles the definition of learning in psychology, where learning is any change in the subject's behavior to repeated situation, after reducing other factory . Automated learning should be defined as the capability of an intelligent system to machine improve its behavior (or its performance) as a result of its previous experience, two outstanding of automated learning are:

- Concept learning
- Inductive learning, or learning by example

1.6. Artificial Intelligence Problems

The most natural definition of AI problem would be that the fundamental AI problem is problem solving itself, from the very beginning AI pioneers have attempted to solve the problem of automated game-playing, and later the problem of automated of reasoning and theorem proving, these problem are now viewed as problem internal to AI, not much work is presently begin done in this area.
Today the genuine AI problem originate from the quest and the efforts of scientists and engineering to develop new method for solving existing ,as well as new problem ,this is an inevitable occurrence with the perpetual technological progress we are witnessing, the associated problem to be solved, requiring the method of intelligence can be classified as problem of:

- Natural language processing
- Pattern recognition
- Computer robotics
- Expert system

1.7. Summary

Artificial intelligent system, in this title of our chapter to many people asking there self some questions are we concerned with thinking or behavior? Do we want to model humans; it's possible and many think from that way.

In this chapter we have defined and established AI background, by considering the meaning of AI history and problems

And also we had considering the solution that it was funded for that by using AI progress theoretical basis method that understanding the theoretical basis for intelligence has gone hand in hand with improvements in the capabilities of real system.

{ "The scientific understanding of the mechanisms underlying thought and intelligent behavior and their embodiment in machines " by John McCarthy1956}.

Chapter TWO

EXPERT SYSTEM

2.1. Overview

So far we have talked a lot about how we can represent knowledge, but not so much about how we can use it to solve real practical problems. This chapter will therefore look at how some of the techniques discussed so far are used in expert system- systems that provide expert quality advice, diagnoses and recommendations given real world problems.

2.2. Introduction of expert system

Expert systems are meant to solve real problems, which normally would require a specialized human expert (such as a doctor or a mineralogist). Building an expert system therefore first involves extracting the relevant knowledge from the human expert. Such knowledge is often heuristic in nature, based on useful ``rules of thumb" rather than absolute certainties. Extracting it from the expert in a way that can be used by a computer is generally a difficult task, requiring its own expertise. A knowledge engineer has the job of extracting this knowledge and building the expert system knowledge base. A first attempt at building an expert system is unlikely to be very successful. This is partly because the expert generally finds it very difficult to express exactly what knowledge and rules they use to solve a problem. Much of it is almost subconscious, or appears so obvious they don't even bother mentioning it. Knowledge acquisition for expert systems is a big area of research, with a wide variety of techniques developed. However, generally it is important to develop an initial prototype based on information extracted by interviewing the expert, and then iteratively refine it based on feedback both from the expert and from potential users of the expert system, Expert systems have been used to solve a wide range of problems in domains such as medicine, mathematics, engineering, geology, computer science, business, law, defense and education. Within each domain, they have been used to solve problems of different types. Types of problem involve diagnosis e.g., of a system fault, disease or student error; design of a computer systems, hotel etc; and interpretation of, for example, geological data. The appropriate problem solving technique tends to depend more on the problem type than on the domain. Whole books have been written on how to choose your knowledge representation and reasoning methods given characteristics of your problem.

2.3. What is expert system?

One of the results of research in the area of artificial intelligence has been the development of techniques, which allow the modeling of information at higher levels of abstraction. These techniques are embodied in languages or tools, which allow programs to be built that closely, resemble human logic in their implementation and are therefore easier to develop and maintain. These programs, which emulate human expertise in well-defined problem domains, are called expert systems. The availability of expert system tools, such as CLIPS, has greatly reduced the effort and cost involved in developing an expert system. Rule-based programming is one of the most commonly used techniques for developing expert systems. In this programming paradigm, rules are used to represent heuristics, or "rules of thumb," which specify a set of actions to be performed for a given situation. A rule is composed of an if portion and a then portion. The if portion of a rule is a series of patterns which specify the facts (or data) which cause the rule to be applicable. The process of matching facts to patterns is called pattern matching. The expert system tool provides a mechanism, called the inference engine, which automatically matches facts against patterns and determines which rules are applicable. The if portion of a rule can actually be thought of as the whenever portion of a rule since pattern matching always occurs whenever changes are made to facts. The then portion of a rule is the set of actions to be executed when the rule is applicable. The actions of applicable rules are executed when the inference engine is instructed to begin execution. The inference engine selects a rule and then the actions of the selected rule are executed (which may affect the list of applicable rules by adding or removing facts). The inference engine then selects another rule and executes its actions. This process continues until no applicable rules remain

2.4. Expert system definitions

Definitions of expert systems vary. Some definitions are based on function. Some definitions are based on structure. Some definitions have both functional and structural components. Many early definitions assume rule-based reasoning

Functional Components:

what the system does (rather than how) "a computer program that behaves like a human expert in some useful ways." [Winston & Prendergast, 1984, p.6] [8]

Problem area:

"solve problems efficiently and effectively in a narrow problem area." [Waterman, 1986, p.xvii]

" Typically, pertains to problems that can be symbolically represented" [Liebowitz, 1988, p.3] [9]

Problem difficulty:

" apply expert knowledge to difficult real world problems" [Waterman, 1986, p.18]

" solve problems that are difficult enough to require significant human expertise for their solution" [Edward Feigenbaum in Harmon & King, 1985, p.5]

" address problems normally thought to require human specialists for their solution" [Michaelsen et al, 1985, p. 303]. [10]

Performance requirement:

"the ability to perform at the level of an expert" [Liebowitz, 1988, p.3]

" programs that mimic the advice-giving capabilities of human experts." [Brule, 1986, p.6]

"Matches a competent level of human expertise in a particular field." [Bishop, 1986, p.38]

"Can offer intelligent advice or make an intelligent decision about a processing function." [British Computer Society's Specialist Group in Forsyth, 1984, pp.9-10]

"Allows a user to access this expertise in a way similar to that in which he might consult a human expert, with a similar result." [Edwards and Connell, 1989, p.3] [11]

Explain reasoning

"the capability of the system, on demand, to justify its own line of reasoning in a manner directly intelligible to the enquirer." [British Computer Society's Specialist Group in Forsyth, 1984, p.9-10]

"Incorporation of explanation processes" [Liebowitz, 1988, p.3][12]

"Expert systems are computer programs mimicking the decision-making processes of humans in a limited area of expertise." (Morgan, 1997). Library applications of expert systems typically include a logical question and answer process or series of menus, a matching of user answers with appropriate information sources, a list of recommended sources, and, in some cases, a way to redirect users after mistakes

2.5. Definition Expert System Building Tools

An expert system tool, or shell, is a software development environment containing the basic components of expert systems. Associated with a shell is a prescribed method for building applications by configuring and instantiating these components. Some of the generic components of a shell are shown in Figure 2.1and described below. The core components of expert systems are the knowledge base and the reasoning engine.

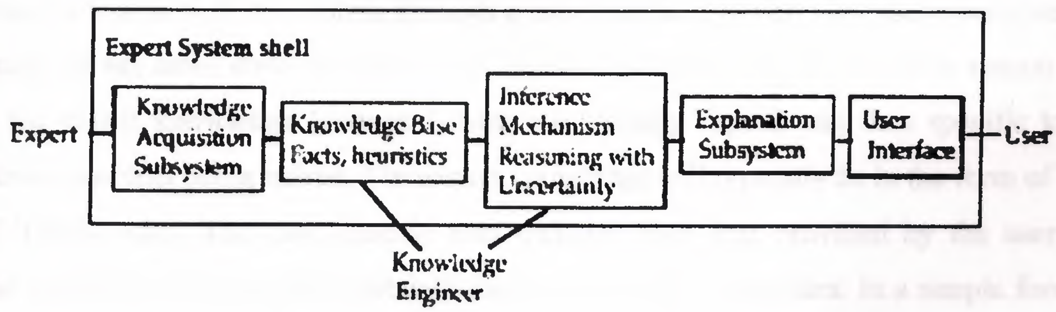


Figure 2.1. Basic Components of Expert System Tools

Knowledge base: A store of factual and heuristic knowledge. An ES tool provides one or more knowledge representation schemes for expressing knowledge about the application domain. Some tools use both frames (objects) and IF-THEN rules. In PROLOG the knowledge is represented as logical statements.

Reasoning engine: Inference mechanisms for manipulating the symbolic information and knowledge in the knowledge base to form a line of reasoning in solving a problem. The inference mechanism can range from simple modus pone's backward chaining of IF-THEN rules to case-based reasoning.

Knowledge acquisition subsystem: A subsystem to help experts build knowledge bases. Collecting knowledge needed to solve problems and build the knowledge base continues to be the biggest bottleneck in building expert systems.

Explanation subsystem: A subsystem that explains the system's actions. The explanation can range from how the final or intermediate solutions were arrived at to justifying the need for additional data.

User interface: The means of communication with the user. The user interface is generally not a part of the ES technology, and was not given much attention in the past. However, it is now widely accepted that the user interface can make a critical difference in the perceived utility of a system regardless of the system's performance.

2.6. Expert System Architecture

The user interacts with the system through a *user* interface, which may use menus, natural language or any other style of interaction). Then an inference engine is used to reason with both the expert knowledge (extracted from our friendly expert) and data specific to the particular problem being solved. The expert knowledge will typically be in the form of a set of IF-THEN rules. The case specific data includes both data provided by the user and partial conclusions (along with certainty measures) based on this data. In a simple forward chaining rule-based system the case specific data will be the elements in working memory.

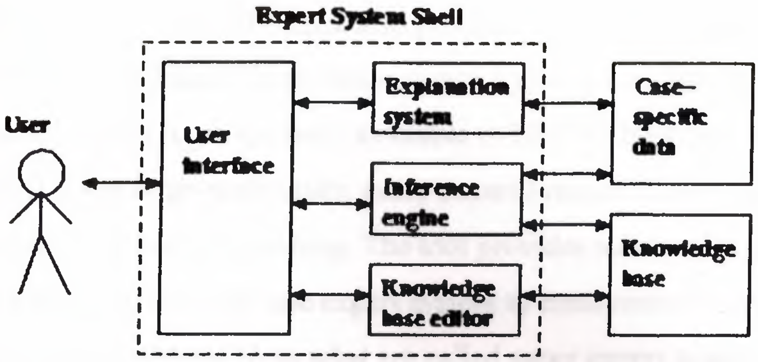


Figure 2.2 most important modules that make up

A rule-based expert system

One important feature of expert systems is the way they (usually) separate domain specific knowledge from more general-purpose reasoning and representation techniques. The general-purpose bit (in the dotted box in the figure) is referred to as an *expert system shell.* As we see in the figure, the shell will provide the inference engine (and knowledge representation scheme), a user interface, an explanation system and sometimes a knowledge base editor. Given a new kind of problem to solve (say, car design), we can usually find a shell that provides the right sort of support for that problem, so all we need to do is provide the expert knowledge. There are numerous commercial expert system shells, each one appropriate for a slightly different range of problems. (Expert systems work in industry includes both writing expert system shells and writing expert systems using shells.) Using shells to write expert systems generally greatly reduces the cost and time of development (compared with writing the expert system from scratch).

2.7. Profile of a Tool: ES/KERNEL2

ES/KERNEL2, the new version of the current best-selling tool, is geared to the development of large-scale applications It gives the application developer's choice in the use of reasoning methods: rule-based reasoning, object-oriented reasoning, and assumption-based reasoning can all be used within a single expert system. Associated with each reasoning method is a knowledge representation scheme best suited to it? For example, for object-oriented reasoning, knowledge is represented as frames, slots, and methods as it shown in Figure 2.3. ES/KERNEL2 also provides some advanced capabilities such as ATMS (Assumption-based Truth Maintenance System) and case-based reasoning (under development). Fuzzy logic has been available in ES/KERNEL and will be a part of ES/KERNEL2.For large-scale tasks; many expert systems can be connected to perform multi-layered cooperative reasoning. The tool provides a means, in the form of a blackboard data structure, for one expert system to communicate with another. The cooperating expert systems form what are called super expert systems, which in turn can cooperate with each other to solve still larger problems.

2.7.1. Seminal characteristics of ES/KERNEL2 include:

- In place of a knowledge acquisition system intended for experts' use ES/KERNEL2 provides other tools, such as a knowledge editor, to help knowledge engineers enter and modify the knowledge base.
- It provides graphic, as well as multi-media, tools for building the end-user interface. These functionalities are built with X-windows.
- The interface to external databases is designed to allow general-purpose database software to be entered into the system as frames and used in the reasoning process. Conversely, the results of the reasoning process can be stored in the database.

One objective of the ES/KERNEL environment is ease of use. For example, knowledge can be expressed in English or Japanese. If the user wants to know language specifications or grammar while editing, an explanation of a particular term and usage examples can be displayed. Reportedly, more than 50 percent of an ES developer's time is spent developing the end-user graphic interface. ES/KERNEL2 provides a variety of graphic templates and edits functions for the development of the interface.

Another objective is efficiency. A translator converts knowledge into an easy-to-process intermediate language during development, and for the production version a compiler converts the developed knowledge into a format executable at high speed. An extended RETE algorithm matches rules and objects to speed up production system inference. Other features, such as incremental compilation and knowledge partitioning, also save development time.

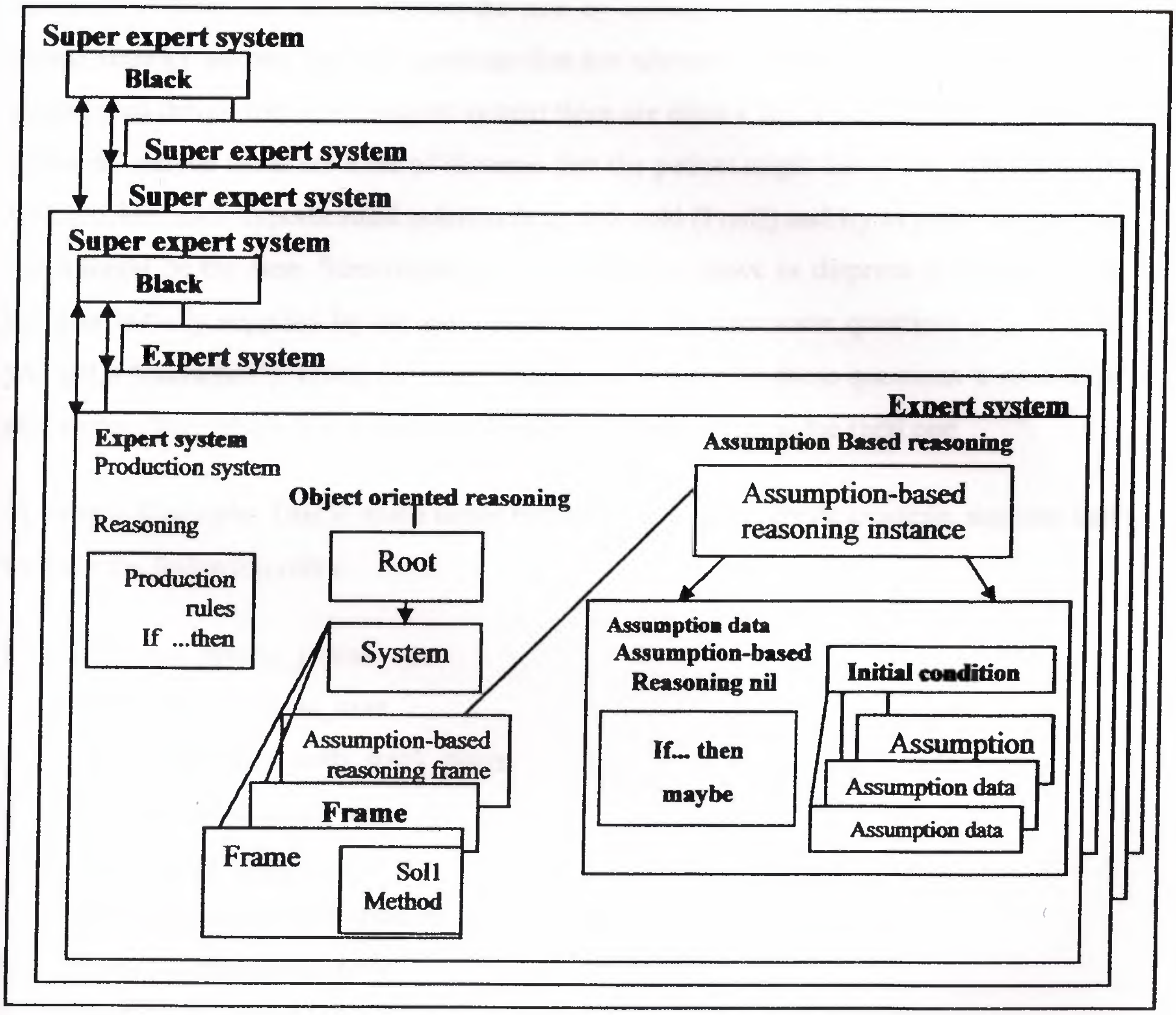


Figure 2.3. ES/KERNEL2, A Tool for Building Multi-Layered, Cooperative Expert Systems

2.8. Rules and Expert Systems

In this section we will show how expert systems based on IF-THEN rules work, and present a very simple expert system shell in Prolog, Rule-based systems can be either goal driven using backward chaining to test whether some hypothesis is true, or data driven, using forward chaining to draw new conclusions from existing data. Expert systems may use either or both strategies, but the most common is probably the goal driven/backward chaining strategy. One reason for this is that normally an expert system will have to collect information about the problem from the user by asking those questions - by using a goal driven strategy we can just ask questions that are relevant to a hypothesized solution. In a simple goal-driven rule-based expert system there are often a set of possible solutions to the problem - maybe these are a set of illnesses that the patient might have. The expert system will consider each hypothesized solution (e.g., has cold (Fred)) and try to prove whether or not it might be the case. Sometimes it won't be able to prove or disprove something from the data initially supplied by the user, so it will ask the user some questions (e.g., ``have you got a headache?"). Using any initial data plus answers to these questions it should be able to conclude which of the possible solutions to the problem is the right one

A Simple Example: This is much better explained through a simple example, suppose that we have the following rules:

1. IF engine_getting_petrol
   AND engine_turns_over
   THEN problem_with_spark_plugs
2. IF NOT engine_turns_over
   AND NOT lights_come_on
   THEN problem_with_battery
3. IF NOT engine_turns_over
   AND lights_come_on
   THEN problem_with_starter
4. IF petrol_in_fuel_tank..............THEN engine_getting_petrol

Our problem is to work out what's wrong with our car given some observable symptoms. There are three possible problems with the car: problem_with_spark_plugs, problem_with_battery, problem_with_starter. We'll assume that we have been provided with no initial facts about the observable symptoms, in the simplest goal-directed system we would try to prove each hypothesized problem (with the car) in turn. First the system would try to prove ``problem_with_spark_plugs". Rule 1 is potentially useful, so the system would set the new goals of proving ``engine_getting_petrol" and ``engine_turns_over". Trying to prove the first of these, rule 4 can be used; with new goal of proving ``petrol_in_fuel_tank" there are no rules, which conclude this (and the system doesn't already know the answer), so the system will ask the user:

Is it true that there's petrol in the fuel tank?

Let's say that the answer is yes. This answer would be recorded, so that the user doesn't get asked the same question again. Anyway, the system now has proved that the engine is getting petrol, so now wants to find out if the engine turns over. As the system doesn't yet know whether this is the case, and as there are no rules, which conclude this, the user will be asked:

Is it true that the engine turns over?

Lets say this time the answer is no. There are no other rules which can be used to prove ``problem_with_spark_plugs" so the system will conclude that this is not the solution to the problem, and will consider the next hypothesis: problem_with_battery. It is true that the engine does not turn over (the user has just said that), so all it has to prove is that the lights don't come one. It will ask the user

Is it true that the lights come on?

Suppose the answer is no. It has now proved that the problem is with the battery. Some systems might stop there, but usually there might be more than one solution, (e.g., more than one fault with the car), or it will be uncertain which of various solutions is the right one. So usually all hypotheses are considered. It will try to prove ``problem_with_starter", but given the existing data (the lights come on) the proof will fail, so the system will conclude that the problem is with the battery. A complete interaction with our very simple system might be:

System: Is it true that there's petrol in the fuel tank?

User: Yes.

System: Is it true that the engine turns over?

User: No.

System: Is it true that the lights come on?

User: No.

System: I conclude that there is a problem with battery.

Note that in general, solving problems using backward chaining involves searching through all the possible ways of proving the hypothesis, systematically checking each of them. A common way of doing this search is the same as in Prolog - depth first search with backtracking.

2.9. Advantages and Disadvantages of Expert Systems

In this section of our booklet we present some of the advantages and disadvantages of existing expert systems.

2.9.1. Advantages of Expert Systems

Permanence - Expert systems do not forget, but human experts may.

Reproducibility - Many copies of an expert system can be made, but training new Human expert is time-consuming and expensive.

Efficiency -can increase throughput and decrease personnel costs.

Although expert systems are expensive to build and maintain, they are inexpensive to operate.

Development and maintenance costs can be spread over many users.

The overall cost can be quite reasonable when compared to expensive and scarce human experts.

Cost savings:

Wages - (elimination of a room full of clerks) other costs - (minimize loan loss)

Consistency – With expert systems similar transactions handled in the same way. This system will make comparable recommendations for like situation. Humans are influenced by:

Decency effects (most recent information having disproportionate impact)

Primacy effects (early information dominates the judgment).

Documentation - An expert system can provide permanent documentation of the decision process.

Completeness - An expert system can review all the transactions, a human expert can only review a sample.

Breadth - The knowledge of multiple human experts can be combined to give a system more breadth that a single person is likely to achieve.

Reduce risk of doing business Consistency of decision-making.

Documentation.

Achieve expertise.

Entry barriers - Expert systems can help a firm create entry barriers for potential competitors.

Differentiation - In some cases, an expert system can differentiate a product or can be related to the focus of the firm (XCON).

Computer programs are best in those situations where there is a structure that is noted as previously existing or can be elicited.

2.9.2. Disadvantages of Rule-Based Expert Systems

Common sense - In addition to a great deal of technical knowledge, human experts have common sense. It is not known how to give expert systems common sense.

Creativity - Human experts can respond creatively to unusual situations, expert systems cannot

Learning - Human experts automatically adapt to changing environments; expert systems must be explicitly updated. Case-based reasoning and neural networks are methods that can incorporate learning.

Sensory Experience - Human experts have available to them a wide range of sensory experience; expert systems are currently dependent on symbolic input.

Degradation - Expert systems are not good at recognizing when no answer exists or when the problem is outside their area of expertise.

2.10. Summary

In this chapter we have give on explain of Expert systems, Expert systems are computer programs mimicking the decision-making processes of humans in a limited area of expertise. (Morgan, 1997). Library applications of expert systems typically include a logical question and answer process or series of menus, a matching of user answers with appropriate information sources, a list of recommended sources, and, in some cases, a way to redirect users after mistakes.

Most expert systems are developed via specialized software tools called shells. These shells come equipped with an inference mechanism (backward chaining, forward chaining, or both), and require knowledge to be entered according to a specified format (all of which might lead some to categorize OPS5 as a shell). They typically come with a number of other features, such as tools for writing hypertext, for constructing friendly user interfaces, for manipulating lists, strings, and objects, and for interfacing with external programs and databases.

CHAPTER THREE

FUZZY SYSTEM

3.1. Overview

Formal control logic is based in the teachings of Aristotle, where an element either is or is not a member of a particular set. Since many of the objects encountered in the real world do not fall into precisely defined membership criteria and in this chapter of our project we hade explain and find the most particular science method to performing such problem.

3.2. Introduction to fuzzy system

Many decision-making and problem-solving tasks are too complex to be understood quantitatively, however, people succeed by using knowledge that is imprecise rather than precise. Fuzzy set theory, originally introduced by Lotfi Zadeh in the 1960's, resembles human reasoning in its use of approximate information and uncertainty to generate decisions. It was specifically designed to mathematically represent uncertainty and vagueness and provide formalized tools for dealing with the imprecision intrinsic to many problems. By contrast, traditional computing demands precision down to each bit. Since knowledge can be expressed in a more natural by using fuzzy sets, many engineering and decision problems can be greatly simplified.

Fuzzy set theory implements classes or groupings of data with boundaries that are not sharply defined (i.e., fuzzy). Any methodology or theory implementing "crisp" definitions such as classical set theory, arithmetic, and programming, may be "justified" by generalizing the concept of a crisp set to a fuzzy set with blurred boundaries. The benefit of extending crisp theory and analysis methods to fuzzy techniques is the strength in solving real-world problems, which inevitably entail some degree of imprecision and noise in the variables and parameters measured and processed for the application. Accordingly, linguistic variables are a critical aspect of some fuzzy logic applications, where general terms such a "large," "medium," and "small" are each used to capture a range of numerical values. While similar to conventional quantization, fuzzy logic allows these stratified sets to overlap (e.g., a 85 kilogram man may be classified in both the "large" and "medium" categories, with varying degrees of belonging or membership to each group). Fuzzy set theory encompasses fuzzy logic, fuzzy arithmetic, fuzzy mathematical programming, fuzzy topology, fuzzy graph theory, and fuzzy data analysis, though the term fuzzy logic is often used to describe all of these.

Fuzzy logic emerged into the mainstream of information technology in the late 1980's and early 1990's. Fuzzy logic is a departure from classical Boolean logic in that it implements soft linguistic variables on a continuous range of truth values which allows intermediate values to be defined between conventional binary. It can often be considered a superset of Boolean or "crisp logic" in the way fuzzy set theory is a superset of conventional set theory. Since fuzzy logic can handle approximate information in a systematic way, it is ideal for controlling nonlinear systems and for modeling complex systems where an inexact model exists or systems where ambiguity or vagueness is common. A typical fuzzy system consists of a rule base, membership functions, and an inference procedure. Today, fuzzy logic is found in a variety of control applications including chemical process control, manufacturing, and in such consumer products as washing machines, video cameras, and automobiles. [13]

Fuzzy logic starts with and builds on a set of user-supplied human language rules. The fuzzy systems convert these rules to their mathematical equivalents. This simplifies the job of the system designer and the computer, and results in much more accurate representations of the way systems behave in the real world.

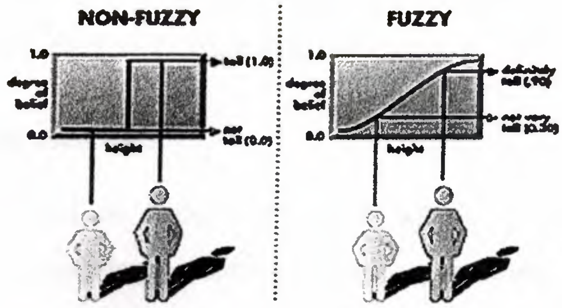


Figure 3.1. A set of user-supplied human language

3.3. What Is Fuzzy Logic?

Fuzzy logic is a powerful problem-solving methodology with a myriad of applications in embedded control and information processing. Fuzzy provides a remarkably simple way to draw definite conclusions from vague, ambiguous or imprecise information. In a sense, fuzzy logic resembles human decision making with its ability to work from approximate data and find precise solutions.

Unlike classical logic which requires a deep understanding of a system, exact equations, and precise numeric values, Fuzzy logic incorporates an alternative way of thinking, which allows modeling complex systems using a higher level of abstraction originating from our knowledge and experience. Fuzzy Logic allows expressing this knowledge with subjective concepts such as very hot, bright red, and a long time which are mapped into exact numeric ranges.

Fuzzy Logic has been gaining increasing acceptance during the past few years. There are over two thousand commercially available products using Fuzzy Logic, ranging from washing machines to high speed trains. Nearly every application can potentially realize some of the benefits of Fuzzy Logic, such as performance, simplicity, lower cost, and productivity.

Fuzzy Logic has been found to be very suitable for embedded control applications. Several manufacturers in the automotive industry are using fuzzy technology to improve quality and reduce development time. In aerospace, fuzzy enables very complex real time problems to be tackled using a simple approach. In consumer electronics, fuzzy improves time to market and helps reduce costs. In manufacturing, fuzzy is proven to be invaluable in increasing equipment efficiency and diagnosing malfunctions.

3.3.1 What is Fuzzy Logic used for?

Fuzzy logic has been used in fuzzy controllers which are widely used in control applications including refrigerators, washing machines, welding machines, cameras and robots; Fault and failure diagnosis, image processing, pattern classifying, traffic problems, collision avoidance, decision support, project planning, fraud detection and in conjunction with neural nets and expert systems.

3.3.2. Who uses Fuzzy Logic?

The Japanese use fuzzy logic controllers widely in their consumer goods. Electrical, mechanical and process engineers, equipment designers, managers, planners, data base designers, neural network users.

3.3.3. Why is Fuzzy Logic Better?

Fuzzy logic is an extension of Boolean logic into the real world where many events are more accurately described by continuous logic.

3.4 Fuzzy Sets

Fuzzy Set Theory was formalized by Professor Lofti Zadeh at the University of California in 1965. What Zadeh proposed is very much a paradigm shift that first gained acceptance in the Far East and its successful application has ensured its adoption around the world.

A paradigm is a set of rules and regulations which defines boundaries and tells us what to do to be successful in solving problems within these boundaries. For example the use of transistors instead of vacuum tubes is a paradigm shift - likewise the development of Fuzzy Set Theory from conventional bivalent set theory is a paradigm shift. Bivalent Set Theory can be somewhat limiting if we wish to describe a 'humanistic' problem mathematically. For example, Figure3.2 below illustrates bivalent sets to characterize the temperature of a room.

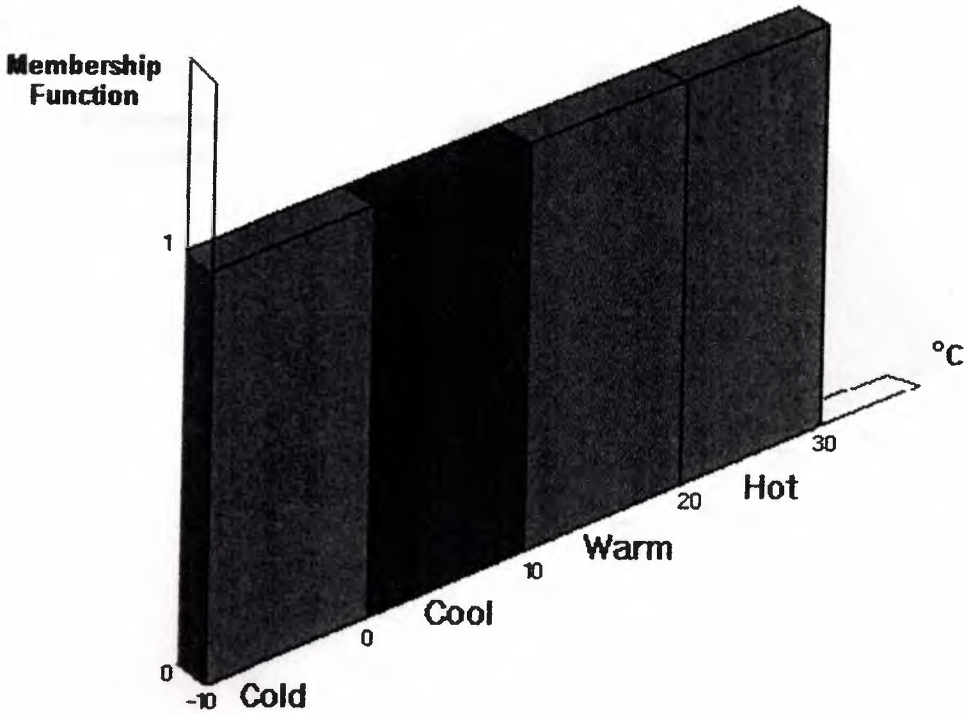


Figure 3.2. bivalent sets to character the tempt .of a room

The most obvious limiting feature of bivalent sets that can be seen clearly from the diagram is that they are mutually exclusive - it is not possible to have membership of more than one set (opinion would widely vary as to whether 50 degrees Fahrenheit is 'cold' or 'cool' hence the expert knowledge we need to define our system is mathematically at odds with the humanistic world). Clearly, it is not accurate to define a transition from a quantity such as 'warm' to 'hot' by the application of one degree Fahrenheit of heat. In the real world a smooth (unnoticeable) drift from warm to hot would occur. This natural phenomenon can be described more accurately by Fuzzy Set Theory. Figure3.3. below shows how fuzzy sets quantifying the same information can describe this natural drift.

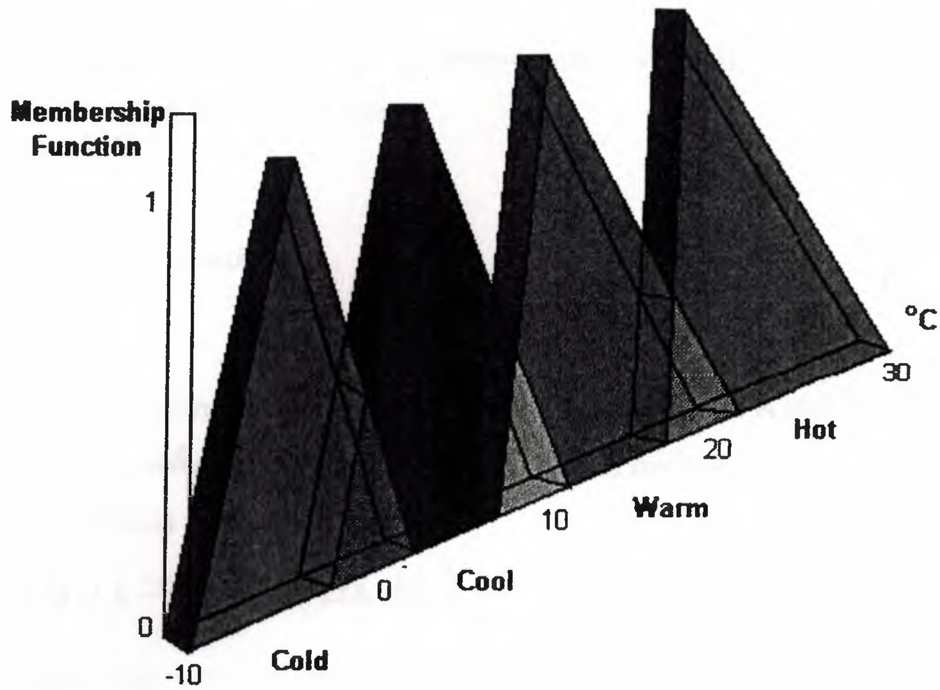


Figure 3.3. Fuzzy sets to characterize the temp of a room

3.4.1. Definitions.

- Universe of Discourse

  The Universe of Discourse is the range of all possible values for an input to a fuzzy system.

- Fuzzy Set

  A Fuzzy Set is any set that allows its members to have different grades of membership (membership function) in the interval [0, 1].

- Support

  The Support of a fuzzy set F is the crisp set of all points in the Universe of Discourse U such that the membership function of F is non-zero.

- Crossover point

  The Crossover point of a fuzzy set is the element in U at which its membership function is 0.5.

- Fuzzy Singleton

  A Fuzzy singleton is a fuzzy set whose support is a single point in U with a membership function of one

3.4.2 Fuzzy Set Operations.

- Union

  The membership function of the Union of two fuzzy sets A and B with membership functions μ_A and μ_B respectively is defined as the maximum of the two individual membership functions as in fig.3.3

$$\mu_{A \cup B} = \max(\mu_A, \mu_B)$$

  The Union operation in Fuzzy set theory is the equivalent of the OR operation in Boolean algebra.

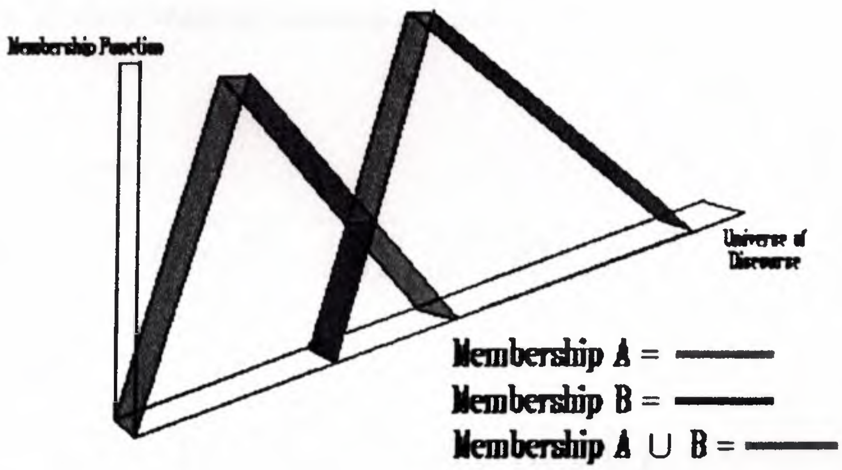


Figure 3.4. The membership function of the Union of two fuzzy sets

- **Complement**

The membership function of the Complement of a Fuzzy set A with membership function μ_A is defined as show in fig 3.4

$$\mu_{\bar{A}} = 1 - \mu_A$$

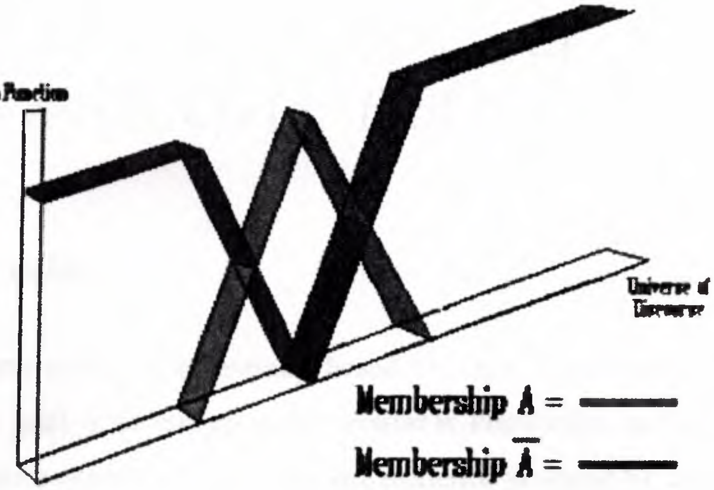


Figure 3.5. The membership function of the Complement of a Fuzzy

The following rules which are common in classical set theory also apply to Fuzzy set theory.

- De Morgan's law

$$\overline{(A\cap B)} = \bar{A}\cap\bar{B}\,,\ \overline{(A\cup B)} = \bar{A}\cap\bar{B}$$

- Associatively

$$(A\cup B)\cup C = A\cup(B\cup C)$$
$$(A\cap B)\cap C = A\cap(B\cap C)$$

- Commutatively

$$A\cap B = B\cap A\,,\ A\cup B = B\cup A$$

▪ Distributives

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$
$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

3.5. Generating fuzzy rules

The construction of a fuzzy model is essentially based on data and/or expertise on the system. In both cases our goal is to compress the available knowledge and/or data in a manner that enables us to state general assertions on the evidence provided by simple, small and comprehensible knowledge bases.

If empiric data is available, we should aim to using a sample which well represents the re-sportive population. The sample size depends on the rule generation algorithm, the size of the population, the homogeneity of the data, the dimensionality of the model space, the objective of the model construction and the established error limits of the system, inter alias.

For example, when generating the rules, the *grid partition technique*, which generates rules by using all the combinations of input and output values, usually requires large sample sizes and heavy computation. If a one-input-one-output system requires at least ten data points, a four-variable system should use 10^3=1000 data points. It also would yield large rule bases if several variables are included in the system. Four variables, each using three values, already yield 34=81 combinations (i.e., system) as it in figure (3.5, 6)

If the sample size is sufficiently large, we may overcome several of the preceding problems by dividing the data into two parts, training data, and control data. The model construction and its possible tuning are based on the former set, whereas we assess this model by testing it with the latter set (and vice versa, if necessary). Then, our model can yield good outputs on a general level. A typical over determination problem occurs when the model fits the training data well, but clearly yields unsatisfactory outputs for the control data. We must also bear in mind that these models are only appropriate to interpolation, whereas extrapolation is problematic to any model.

If appropriate data is unavailable and we only utilize expertise, the system construction may be more problematic from the standpoint of generalization and applicability. We can also utilize expertise to support the data or vice versa. Below the standard normal distribution is depicted. By using the scatter technique (SubclusterC algorithm), we may generate fuzzy cluster centers (rules), and the number the rules is dependent on our goal and problem-setting.

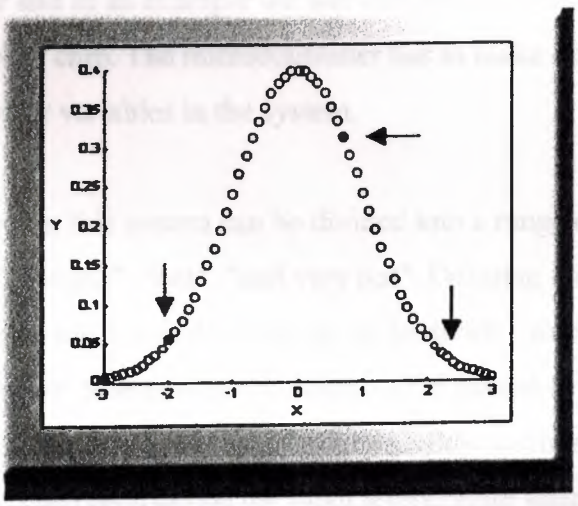


Figure 3.6. Three Fuzzy Cluster Centers (Rules)

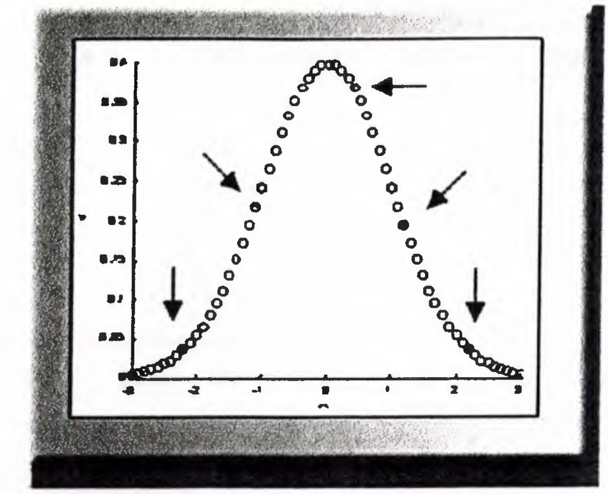


Figure 3.7. Five Fuzzy Cluster Centers (Rules)

3.6. Example of fuzzy modeling

In this part of our chapter and as an example we will consider an antilock braking system, directed by a microcontroller chip. The microcontroller has to make decisions based on brake temperature, speed, and other variables in the system.

The variable "temperature" in this system can be divided into a range of "states", such as: "cold", "cool", "moderate", "warm", "hot", "and very hot". Defining the bounds of these states is a bit tricky. An arbitrary threshold might be set to divide "warm" from "hot", but this would result in a discontinuous change when the input value passed over that threshold. The way around this is to make the states "fuzzy", that is, allow them to change gradually from one state to the next. You could define the input temperature states using "membership functions" such as the following:

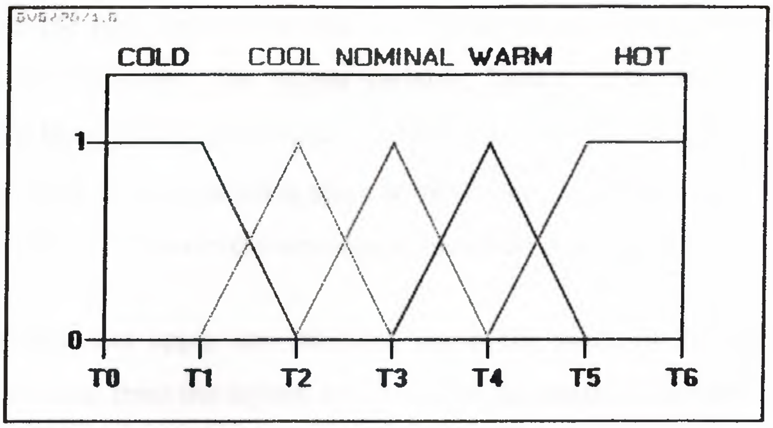


Figure 3.8. Variable "temperature divided into a range of "states

With this scheme, the input variable's state no longer jumps abruptly from one state to the next. Instead, as the temperature changes, it loses value in one membership function while gaining value in the next. At any one time, the "truth value" of the brake temperature will almost always be in some degree part of two membership functions: 0.6 nominal and 0.4 warm, or 0.7 nominal and 0.3 cool, and so on.

The input variables in a fuzzy control system are in general mapped into by sets of membership functions similar to this, known as "fuzzy sets". The process of converting a crisp input value to a fuzzy value is called "fuzzification".

A control system may also have various types of switch, or "ON-OFF", inputs along with its analog inputs, and such switch inputs of course will always have a truth value equal to either 1 or 0, but the scheme can deal with them as simplified fuzzy functions that are either one value or another.

Given "mappings" of input variables into membership functions and truth values, the microcontroller then makes decisions for what action to take based on a set of "rules", each of the form:

```
IF brake temperature IS warm AND speed IS not very fast
THEN brake pressure IS slightly decreased.
```

In this example, the two input variables are "brake temperature" and "speed" that have values defined as fuzzy sets. The output variable, "brake pressure", is also defined by a fuzzy set that can have values like "static", "slightly increased", "slightly decreased", and so on. This rule by itself is very puzzling since it looks like it could be used without bothering with fuzzy logic, but remembers the decision is based on a *set* of rules:

- *All* the rules that apply are invoked, using the membership functions and truth values obtained from the inputs, to determine the result of the rule.
- This result in turn will be mapped into a membership function and truth value controlling the output variable.
- These results are combined to give a specific ("crisp") answer, the actual brake pressure, a procedure known as "defuzzification".

This combination of fuzzy operations and rule-based "inference" describes a "fuzzy expert system", Traditional control systems are based on mathematical models in which the control system is described using one or more differential equations that define the system response to its inputs. Such systems are often implemented as "proportional-integral-derivative (PID)" controllers. They are the products of decades of development and theoretical analysis, and are highly effective.

If PID and other traditional control systems are so well-developed, why bother with fuzzy control? It has some advantages. In many cases, the mathematical model of the control process may not exist, or may be too "expensive" in terms of computer processing power and memory, and a system based on empirical rules may be more effective.

Furthermore, fuzzy logic is well suited to low-cost implementations based on cheap sensors, low-resolution analog-to-digital converters, and 4-bit or 8-bit one-chip microcontroller chips. Such systems can be easily upgraded by adding new rules to improve performance or add new features. In many cases, fuzzy control can be used to improve existing traditional controller systems by adding an extra layer of intelligence to the current control method.

3.7. The fuzzy logic control problem

A typical embedded control problem is usually solved with a controller that follows the concept of Figure 3.8 Naturally, depending on the embedded engineer's experience and budget; this underlying structure may look quite different from one design to another. In all cases, the task of the controller is to modify the Process Inputs so that the Process Outputs converge towards the Set Points. This is usually referred to as the control law, and can be implemented by several control algorithms like PID, Optimal Control, Adaptive Control, FL Control, Neural Network Control, etc.

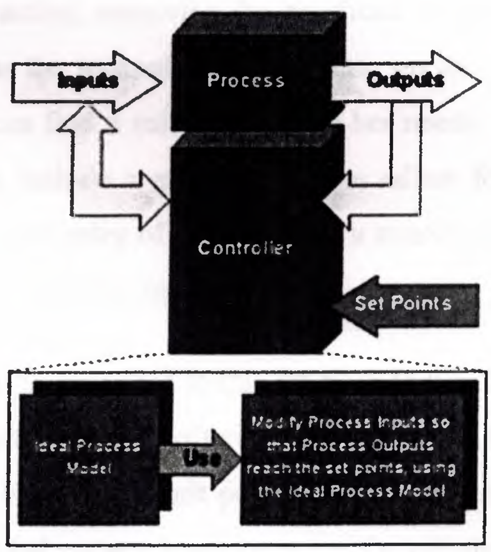


Figure 3.9. The typical controller action

Each control algorithm has advantages and disadvantages, relative to the nature of the process under control; however FL control algorithms can offer significant advantages to the embedded systems engineer, for any system:

- Straight-forward definition of the system variables, using linguistic & graphical methods,
- Rapid modeling of the process model using simple language if...then... statements,
- Easy access to the FL definition of the controller, resulting in easy debugging and tuning,
- Ability to model non-linear processes with the same ease of modeling linear ones,
- Availability of powerful FL development tools with build-in:
  - Source code generators
  - Reporting tools
  - Debugging and tuning tools.

3.8. The fuzzy logic controller development

For the development of a FL controller, it is strongly recommended to use one of the commercially available development tools. These tools address the problem to its appropriate level of abstraction, removing the overhead for one to become a fuzzy logic expert, before she can use the technology. Depending on the embedded system engineer and FL knowledge, one can find a suitable tool for her needs. 's preferences Most of the advanced tools nowadays include a graphical design editor for the definition of the FL variables, a text editor for the entry of the FL rules, a source code generator for exporting the FL design and some type of debugging method.

The battery charge controller described in this article was developed with the Fuzzy Logic Development Environment (FLDE), offered by Syndics Ltd. FLDE is one of the commercial results of the European Esprit project OMI-FEM and is designed to serve best the requirements of embedded system development in the automotive domain. The program runs under MS-Windows and generates self-contained, strict ANSI-C code, suitable for immediate compilation and embedding in an automotive system. The FL development with FLDE is quite straight-forward and is based on a step-by-step definition of three basic entities: the FL project, the FL nouns and the FL rules. The project is the reference point for all the files and entities that constitute a FL system. The nouns are the FL variables, while the rules are the textual sentences that interconnect them.

3.9. Summary

As we defined in the suppliant of this chapter that was not long ago considered another esoteric, scientific branch of the academic world, is now an all pervasive technology with hundreds of applications in control systems and decision support systems.

We have considered in this chapter the logic of how fuzzy is working by considering a small example about it as you have read, and we have considered also important method to make it easy to understand fuzzy logic.

And we have entered to fuzzy modeling by consider the way we have to use in modeling and the steps it have to be started in our modeling ,and also we consider same problem fuzzy logic control and concept action to solved some like of this kinds. And in end it become, the way of how we can develop fuzzy logic

CHAPTER FOUR

NEURAL NETWORKS

4.1. Overview

This chapter is an introduction to Artificial Neural Networks. The various types of neural networks are explained and demonstrated, applications of neural networks "ANNs", and a detailed historical background is provided. The connection between the artificial and the real thing is also investigated and explained. Finally, the mathematical models involved are presented and demonstrated

4.2. Introduction

The power and speed of modern digital computers is truly astounding. No human can ever hope to compute a million operations a second. However, there are some tasks for which even the most powerful computers cannot compete with the human brain, perhaps not even with the intelligence of an earthworm.

Neural Networks approaches this problem by trying to mimic the structure and function of our nervous system. Many researchers believe that Artificial Intelligence and neural networks are completely opposite in their approach. Conventional AI is based on the symbol system hypothesis. Loosely speaking, a symbol system consists of indivisible entities called symbols, which can form more complex entities, by simple rules. The hypothesis then states that such a system is capable of and is necessary for intelligence.

The general belief is that Neural Networks is a sub-symbolic science. Before symbols themselves are recognized, some thing must be done so that conventional AI can then manipulate those symbols. To make this point clear, consider symbols such as cow, grass, house etc. Once these symbols and the "simple rules" which govern them are known, conventional AI can perform miracles. But to discover that something is a cow is not trivial. It can perhaps be done using conventional AI and symbols such as - white, legs, etc. But it would be tedious and certainly, when you see a cow, you instantly recognize it to be so, without counting its legs.

But this belief - that AI and Neural Networks are completely opposite, is not valid because, even when you recognize a cow, it is because of certain properties which you observe, that you conclude that it is a cow. This happens instantly because various parts of the brain function in parallel. All the properties which you observe are "summed up". Certainly there are symbols here and rules - "summing up". The only difference is that in AI, symbols are strictly indivisible, whereas here, the symbols (properties) may occur with varying degrees or intensities

4.3. What is A Neural Networks?

First of all, when we are talking about a neural network, we should more properly say "artificial neural network" (ANN), because that is what we mean most of the time. Biological neural networks are much more complicated than the mathematical models we use for ANNs. But it is customary to be lazy and drop the "A" or the "artificial".

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well.

4.3.1 Some Definitions of a Neural Network

- A neural network is a system composed of many simple processing elements operating in parallel whose function is determined by network structure, connection strengths, and the processing performed at computing elements or nodes.[14]

- A neural network is a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:

1. Knowledge is acquired by the network through a learning process.
2. Interneuron connection strengths known as synaptic weights are used to store the knowledge [15]

4.3.2. Who is concerned with Neural Networks?

Neural Networks are interesting for quite a lot of very dissimilar people:

- Computer scientists want to find out about the properties of non-symbolic information processing with neural nets and about learning systems in general.
- Engineers of many kinds want to exploit the capabilities of neural networks on many areas (e.g. signal processing) to solve their application problems.
- Cognitive scientists view neural networks as a possible apparatus to describe models of thinking and conscience (High-level brain function).
- Neuron-physiologists use neural networks to describe and explore medium-level brain function (e.g. memory, sensory system, matrices).
- Physicists use neural networks to model phenomena in statistical mechanics and for a lot of other tasks.
- Biologists use Neural Networks to interpret nucleotide sequences.
- Philosophers and some other people may also be interested in Neural Networks for various reasons.

4.4. Historical Background of Neural Network

1. First Attempts: There were some initial simulations using formal logic. McCulloch and Pitts (1943) developed models of neural networks based on their understanding of neurology. These models made several assumptions about how neurons worked. Their networks were based on simple neurons which were considered to be binary devices with fixed thresholds. The results of their model were simple logic functions such as "a or b" and "a and b".

Another attempt was by using computer simulations. Two groups (Farley and Clark, 1954; Rochester, Holland, Haibit and Duda, 1956). The first group (IBM reserchers) maintained closed contact with neuroscientists at McGill University. So whenever their models did not work, they consulted the neuroscientists. This interaction established a multidisciplinary trend which continues to the present day. [16]

2. Promising and Emerging Technology: Not only was neuroscience influential in the development of neural networks, but psychologists and engineers also contributed to the progress of neural network simulations. Rosenblatt (1958) stirred considerable interest and activity in the field when he designed and developed the Preceptor. The Perception had three layers with the middle layer known as the association layer. This system could learn to connect or associate a given input to a random output unit.

Another system was the ADALINE (ADAptive Linear Element) which was developed in 1960 by Widrow and Hoff (of Stanford University). The ADALINE was an analogue electronic device made from simple components. The method used for learning was different to that of the Preceptor; it employed the Least-Mean-Squares (LMS) learning rule [17]

3. Period of Frustration & Disrepute: In 1969 Minsky and Papert wrote a book in which they generalized the limitations of single layer Perceptions to multilayered systems. In the book they said: "...our intuitive judgment that the extension (to multilayer systems) is sterile". The significant result of their book was to eliminate funding for research with neural network simulations. The conclusions supported the disenchantment of researchers in the field. As a result, considerable prejudice against this field was activated [18]

4. Innovation: Although public interest and available funding were minimal, several researchers continued working to develop neuromorphically based computational methods for problems such as pattern recognition. During this period several paradigms were generated which modern work continues to enhance.Grossberg's (Steve Grossberg and Gail Carpenter in 1988) influence founded a school of thought which explores resonating algorithms. They developed the ART (Adaptive Resonance Theory) networks based on biologically plausible models. Anderson and Kohonen developed associative techniques independent of each other. Klopf (A. Henry Klopf) in 1972

developed a basis for learning in artificial neurons based on a biological principle for neuronal learning called homeostasis. Werbos (Paul Werbos 1974) developed and used the back-propagation learning method, however several years passed before this approach was popularized. Back-propagation nets are probably the most well known and widely applied of the neural networks today. In essence, the back-propagation net. Is a Preceptor with multiple layers, a different threshold function in the artificial neuron, and a more robust and capable learning rule. Mari (A. Shun-Ichi 1967) was involved with theoretical developments: he published a paper which established a mathematical theory for a learning basis (error-correction method) dealing with adaptive pattern classification. While Fukushima (F. Kunihiko) developed a step wise trained multilayered neural network for interpretation of handwritten characters. The original network was published in 1975 and was called the Cognition.

5. Re-Emergence: Progress during the late 1970s and early 1980s was important to the re-emergence on interest in the neural network field. Several factors influenced this movement. For example, comprehensive books and conferences provided a forum for people in diverse fields with specialized technical languages, and the response to conferences and publications was quite positive. The news media picked up on the increased activity and tutorials helped disseminate the technology. Academic programs appeared and courses were introduced at most major Universities (in US and Europe). Attention is now focused on funding levels throughout Europe, Japan and the US and as this funding becomes available, several new commercial with applications in industry and financial institutions are emerging. [19]

6. Today: Significant progress has been made in the field of neural networks-enough to attract a great deal of attention and fund further research. Advancement beyond current commercial applications appears to be possible, and research is advancing the field on many fronts. Neurally based chips are emerging and applications to complex problems developing. Clearly, today is a period of transition for neural network technology. [20]

4.5. Human and Artificial Neurons

4.5.1. How the Human Brain Learns?

Much is still unknown about how the brain trains itself to process information, so theories abound. In the human brain, a typical neuron collects signals from others through a host of fine structures called dendrites. The neuron sends out spikes of electrical activity through a long, thin stand known as an axon as in figure 4.1, which splits into thousands of branches. At the end of each branch, a structure called a synapse converts the activity from the axon into electrical effects that inhibit or excite activity from the axon into electrical effects that inhibit or excite activity in the connected neurons as in figure 4.2. When a neuron receives excitatory input that is sufficiently large compared with its inhibitory input, it sends a spike of electrical activity down its axon. Learning occurs by changing the effectiveness of the synapses so that the influence of one neuron on another changes

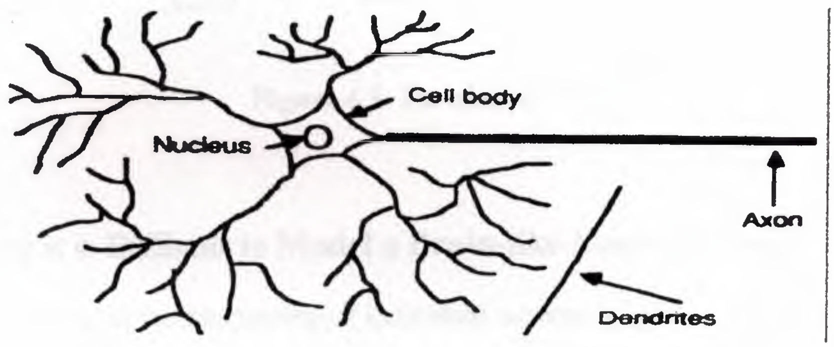


Figure 4.1. Components of a neuron

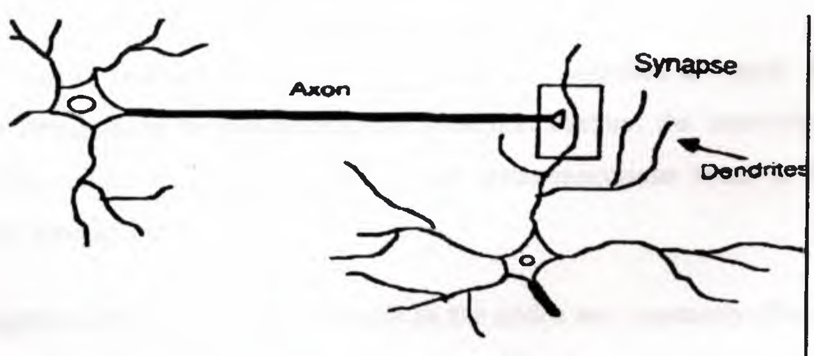


Figure 4.2. the synapse

NEAR EAST UNIVERSITY LIBRARY 1988 - LEFKOŞA

4.5.2. From Human Neurons to Artificial Neurons

We conduct these neural networks by first trying to deduce the essential features of neurons and their interconnections as in figure 4.3. We then typically program a computer to simulate these features. However because our knowledge of neurons is incomplete and our computing power is limited, our models are necessarily gross idealizations of real networks of neurons.

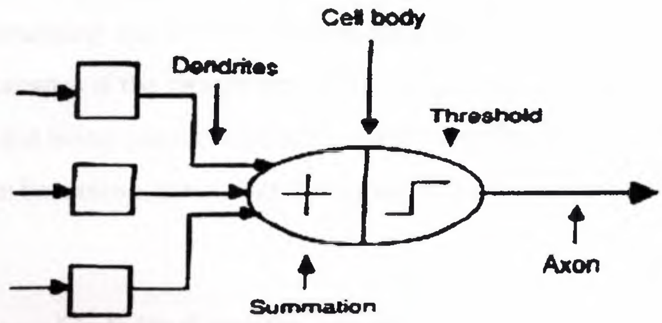


Figure 4.3. The neuron model

4.6. Why it is Difficult to Model a Brain-like Neural Network

We have seen that the functioning of individual neurons is quite simple. Then why is it difficult to achieve our goal of combining the abilities of computers and humans?

The difficulty arises because of the following -

1. It is difficult to find out which neurons should be connected to which. This is the problem of determining the neural network structure. Further, the interconnections in the brain are constantly changing. The initial interconnections seem to be largely governed by genetic factors.

2. The weights on the edges and thresholds in the nodes are constantly changing. This problem has been the subject of much research and has been solved to a large extent. The approach has been as follows -

Given some input, if the neural network makes an error, then it can be determined exactly which neurons were active before the error. Then we can change the weights and thresholds appropriately to reduce this error.

For this approach to work, the neural network must "know" that it has made a mistake. In real life, the mistake usually becomes obvious only after a long time. This situation is more difficult to handle since we may not know which input led to the error.

Also notice that this problem can be considered as a generalization of the previous problem of determining the neural network structure. If this is solved, that is also solved. This is because if the weight between two neurons is zero then, it is as good as the two neurons not being connected at all. So if we can figure out the weights properly, then the structure becomes known. But there may be better methods of determining the structure.

3. The functioning of individual neurons may not be so simple after all. For example, remember that if a neuron receives signals from many neighboring neurons, the combined stimulus may exceed its threshold. Actually, the neuron need not receive all signals at exactly the same time, but must receive them all in a short time-interval.

It is usually assumed that such details will not affect the functioning of the simulated neural network much. But may be it will.

Another example of deviation from normal functioning is that some edges can transmit signals in both directions. Actually, all edges can transmit in both directions, but usually they transmit in only 1 direction, from one neuron to another.

4.7. An engineering approach

4.7.1 A simple neuron

An artificial neuron is a device with many inputs and one output. The neuron has two modes of operation; the training mode and the using mode. In the training mode, the neuron can be trained to fire (or not), for particular input patterns. In the using mode as in figuer4.4, when a taught input pattern is detected at the input, its associated output becomes the current output. If the input pattern does not belong in the taught list of input patterns, the firing rule is used to determine whether to fire or not.

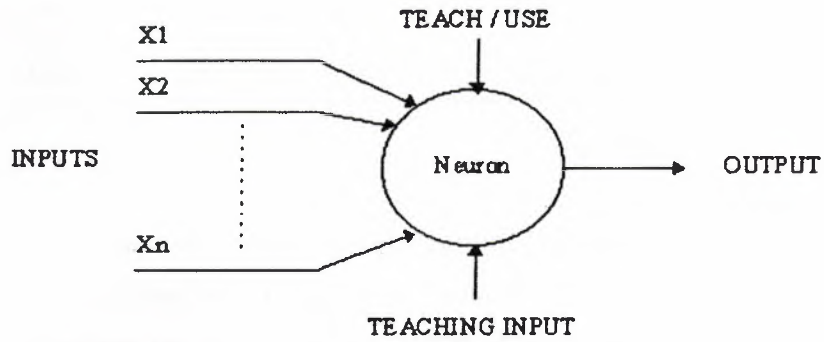


Figure 4.4. A simple artificial neuron

4.7.2. Firing rules

The firing rule is an important concept in neural networks and accounts for their high flexibility. A firing rule determines how one calculates whether a neuron should fire for any input pattern. It relates to all the input patterns, not only the ones on which the node was trained. A simple firing rule can be implemented by using Hamming distance technique. The rule goes as follows:

Take a collection of training patterns for a node, some of which cause it to fire (the 1-taught set of patterns) and others which prevent it from doing so (the 0-taught set). Then the patterns not in the collection cause the node to fire if, on comparison, they have more input elements in common with the 'nearest' pattern in the 1-taught set than with the 'nearest' pattern in the 0-taught set. If there is a tie, then the pattern remains in the undefined state.

For example, a 3-input neuron is taught to output 1 when the input (X1, X2 and X3) is 111 or 101 and to output 0 when the input is 000 or 001. Then, before applying the firing rule, the truth table is;

X1:	0	0	0	0	1	1	1	1
X2:	0	0	1	1	0	0	1	1
X3:	0	1	0	1	0	1	0	1
OUT:	0	0	0/1	0/1	0/1	1	0/1	1

As an example of the way the firing rule is applied, take the pattern 010. It differs from 000 in 1 element, from 001 in 2 elements, from 101 in 3 elements and from 111 in 2 elements. Therefore, the 'nearest' pattern is 000 which belongs in the 0-taught set. Thus the firing rule requires that the neuron should not fire when the input is 001. On the other hand, 011 is equally distant from two taught patterns that have different outputs and thus the output stays undefined (0/1).

By applying the firing in every column the following truth table is obtained;

X1:	0	0	0	0	1	1	1	1
X2:	0	0	1	1	0	0	1	1
X3:	0	1	0	1	0	1	0	1
OUT:	0	0	0	0/1	0/1	1	1	1

The difference between the two truth tables is called the generalization of the neuron. Therefore the firing rule gives the neuron a sense of similarity and enables it to respond 'sensibly' to patterns not seen during training

4.7.3. Pattern Recognition

An important application of neural networks is pattern recognition. Pattern recognition can be implemented by using a feed-forward in figure 4.5 neural network that has been trained accordingly. During training, the network is trained to associate outputs with input patterns. When the network is used, it identifies the input pattern and tries to output the associated output pattern. The power of neural networks comes to life when a pattern that has no output associated with it, is given as an input. In this case, the network gives the output that corresponds to a taught input pattern that is least different from the given pattern.

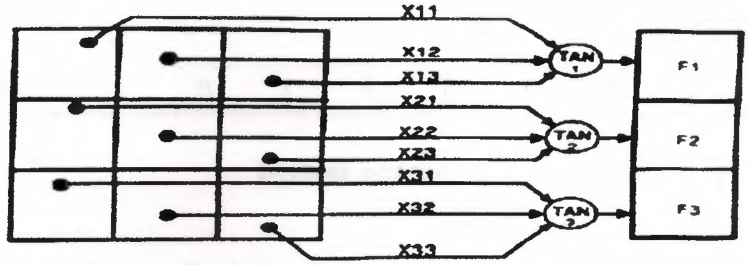


Figure 4.5. Pattern recognition using a feed-forward

And we illustrate this example:

The network of figure 4.5 is trained to recognize the patterns T and H. The associated patterns are all black and all white respectively as shown below.

INPUT OUTPUT INPUT OUTPUT

If we represent black squares with 0 and white squares with 1 then the truth tables for the 3 neurons after generalization are;

X11:	0	0	0	0	1	1	1	1
X12:	0	0	1	1	0	0	1	1
X13:	0	1	0	1	0	1	0	1
OUT:	0	0	1	1	0	0	1	1

Top neuron

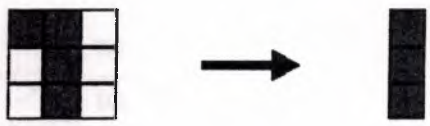
X21:	0	0	0	0	1	1	1	1
X22:	0	0	1	1	0	0	1	1
X23:	0	1	0	1	0	1	0	1
OUT:	1	0/1	1	0/1	0/1	0	0/1	0

Middle neuron

X21:	0	0	0	0	1	1	1	1
X22:	0	0	1	1	0	0	1	1
X23:	0	1	0	1	0	1	0	1
OUT:	1	0	1	1	0	0	1	0

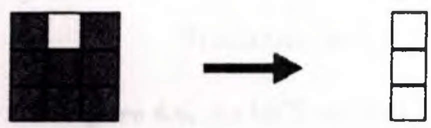
Bottom neuron

From the tables it can be seen the following associations can be extracted:



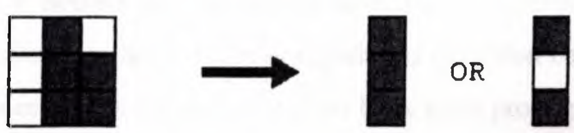
INPUT OUTPUT

In this case, it is obvious that the output should be all blacks since the input pattern is almost the same as the 'T' pattern



INPUT OUTPUT

Here also, it is obvious that the output should be all whites since the input pattern is almost the same as the 'H' pattern.



INPUT OUTPUT

Here, the top row is 2 errors away from the T and 3 from an H. So the top output is black. The middle row is 1 error away from both T and H so the output is random. The bottom row is 1 error away from T and 2 away from H. Therefore the output is black. The total output of the network is still in favor of the T shape.

4.7.4. A more complicated neuron

The previous neuron doesn't do anything that conventional computers don't do already. A more sophisticated neuron (figure 4.6) is the McCulloch and Pitts model (MCP). The difference from the previous model is that the inputs are 'weighted', the effect that each input has at decision making is dependent on the weight of the particular input. The weight of an input is a number which when multiplied with the input gives the weighted input. These weighted inputs are then added together and if they exceed a pre-set threshold value, the neuron fires. In any other case the neuron does not fire.

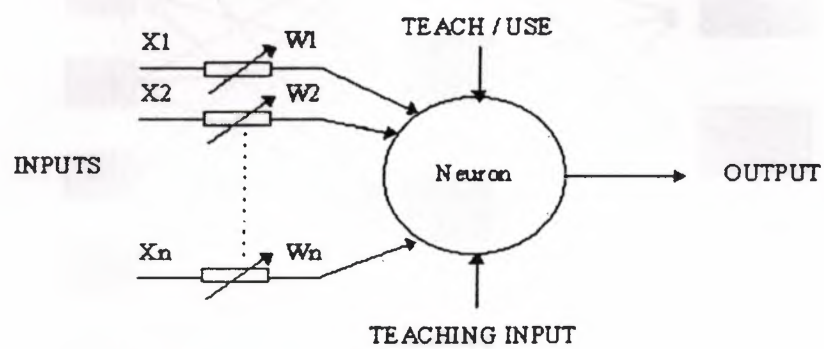


Figure 4.6. An MCP neuron

In mathematical terms, the neuron fires if and only if;

$X1W1 + X2W2 + X3W3 + ... > T$

The addition of input weights and of the threshold makes this neuron a very flexible and powerful one. The MCP neuron has the ability to adapt to a particular situation by changing its weights and/or threshold. Various algorithms exist that cause the neuron to 'adapt'; the most used ones are the Delta rule and the back error propagation. The former is used in feed-forward networks and the latter in feedback networks.

4.8. Architecture of neural networks

4.8.1. Feed-forward networks

Feed-forward ANNs (figure 4.7) allow signals to travel one way only; from input to output. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. Feed-forward ANNs tend to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition. This type of organization is also referred to as bottom-up or top-down.

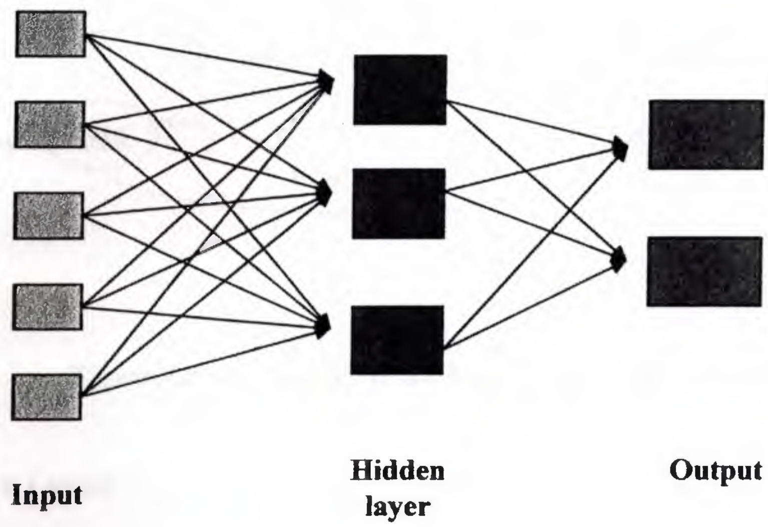


Figure 4.7. An example of a simple feed forward network

4.8.2 Feedback networks

Feedback networks (figure 4.8) can have signals traveling in both directions by introducing loops in the network. Feedback networks are very powerful and can get extremely complicated. Feedback networks are dynamic; their 'state' is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Feedback architectures are also referred to as interactive or recurrent, although the latter term is often used to denote feedback connections in single-layer organizations.

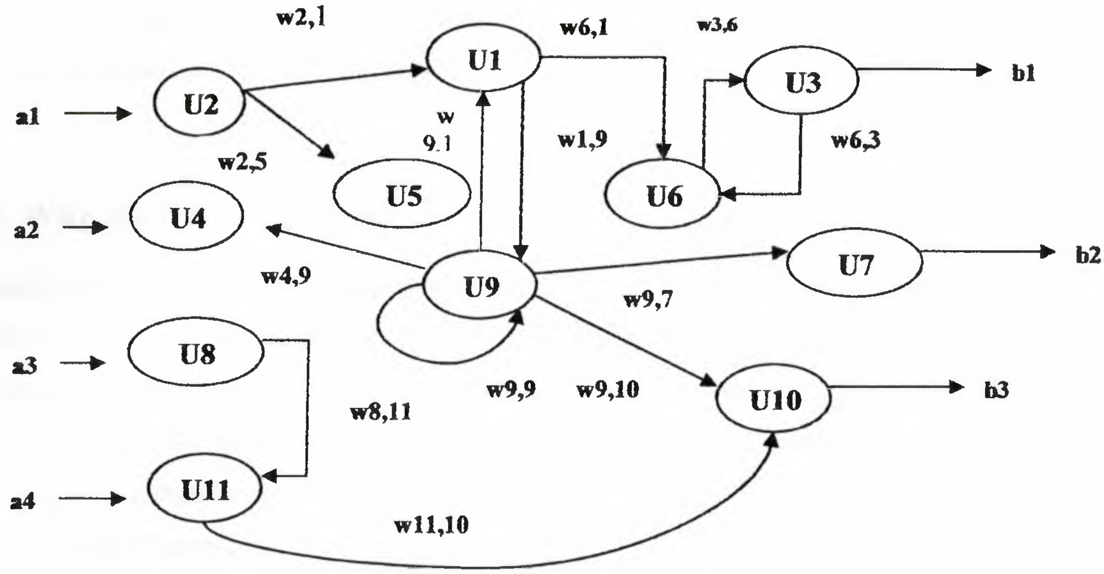


Figure 4.8. An example of a complicated network

4.8.3 Network Layers

The commonest type of artificial neural network consists of three groups, or layers, of units: a layer of "input" units is connected to a layer of "hidden" units, which is connected to a layer of "output" units. As in Figure 4.7

- The activity of the input units represents the raw information that is fed into the network.
- The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units.
- The behavior of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

This simple type of network is interesting because the hidden units are free to construct their own representations of the input. The weights between the input and hidden units determine when each hidden unit is active, and so by modifying these weights, a hidden unit can choose what it represents.

We also distinguish single-layer and multi-layer architectures. The single-layer organization, in which all units are connected to one another, constitutes the most general case and is of more potential computational power than hierarchically structured multi-layer organizations. In multi-layer networks, units are often numbered by layer, instead of following a global numbering.

4.9. Why use Neural Networks?

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest and answer "what if" questions.

Other advantages include:

- Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.

- Self-Organization: An ANN can create its own organization or representation of the information it receives during learning time.

- Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
- The ability to represent any function, linear or not, simple or complicated

Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage

4.10. Summary

The computing world has a lot to gain from neural networks. Their ability to learn by example makes them very flexible and powerful. Furthermore there is no need to devise an algorithm in order to perform a specific task

In this chapter we have demonstrated a basic introduction to neural networks. Within the chapter we have explained that neural networks are groups of select neurons that are connected with one to another.

Also we have explained the definition of artificial neural networks as computing devices that are loosely based on the operation of the brain. Also we have considered the importance of neural networks, who should know about neural networks and their use. We have also explained where neural networks are being used giving some application of there use. Last but not least we have discussed the future of neural networks considering the great deal of neural networks researches going on in worldwide.

CHAPTER FIVE

A.I. APPLICATION

5.1 Overview

This chapter presents monthly stream flow prediction using artificial neural networks (ANN) on mountain watersheds. The procedure addresses the Selection of input variables, the definition of model architecture and the strategy of the learning process.
Results show that spring and summer monthly Stream flows can be adequately represented; improving the results of calculations obtained using other methods. Better stream flow prediction methods should have significant benefits for the optimal use of water resources for irrigation and hydroelectric energy generation.

5.2 Introduction

Several researchers have suggested different mathematical and statistical methods to predict stream flow, approaches such as hydrologic simulation models (Crawford and Linsley, 1962),Snowmelt models (Martinec and Rango, 1992), multiple regression models, transfer functions models (Tripodi, 1999) and empirical models applied to log-transformed flows (Karunanithi et al, 1994) can be mentioned.
A recent study to predict stream flow in the upper Mauled River basin (Fernandez and Tripodi, 1999), illustrates the difficulty of predicting monthly stream flow with reasonable accuracy .The Authors report the importance of incorporating the influence of global phenomena such as ENSO in the hydrologic behavior of watershed in the southern Pacific

Region, Artificial Neural Networks have a structure where non linear functions are present and the parameter identification process is based on techniques which search for global maximums in the space of feasible parameter values, and hence can represent the non linear effects present in the rainfall-runoff processes.

ANN were developed as a information storage models and their parameters are calculated in a manner that resembles natural processes (McCulloch and Pitts, 1943).Details of their properties and the computational process have been presented by Hopfield (1982)and the learning process of ANN is described by Rumel hart and McClelland (1986).

An important advantage of ANN compared to classical stochastic models is that they do not require variables to be stationary and normally distributed (Burke,1991).Non stationary effects present in global phenomena ,in morphological changes in rivers and others can be captured by the inner structure of ANN (Dandy and Maier,1996).Furthermore, ANN are relatively stable with respect to noise in the data and have a good generalization potential to represent input-output relationships.(Zealand et al.1999).[20]

5.3 Artificial neural networks

An ANN is a mathematical model which has a highly connected structure similar to brain cells .They consist of a number of neurons arranged in different layers, an input layer ,an output layer and one or more hidden layers as it in Figure 5.1.

The input neurons receive and process the input signals and send an output signal to other neurons in the network .Each neuron can be connected to the other neurons and has an activation function and a threshold function, which can be continuous, linear or nonlinear functions.

The signal passing through a neuron is transformed by weights which modify the functions and thus the output signal that reaches the following neuron .Modifying the weights for all neurons in the network, changes the output .Once the architecture of the network is defined, weights are calculated so as to represent the desired output through a learning process where the ANN is trained to obtain the expected results, Information available is used to define learning or training data set and a validation data set. Several different architectures and topologies have been proposed for ANN. They differ in terms of architecture, in the learning process and in the training strategy.

A linear model can be represented adequately by a single layer network, while a non linear model is generally associated with a multiple layer network.(LSWC,1999).In this work only three layer net works were considered since it has been shown that they have a good potential to represent linear or non linear outputs.(Minns,1998).According to Elaine and Knight (1996)multiple layer net-works can represent any function ,which implies that the design process has to focus on the definition of the number of neurons and in the learning strategy .Possible model types are feed forward networks (BP1)with back propagation momentum learning process ,hyperbolic activation function for the hidden layer and linear function for the out-put layer;BP2 networks which are the same as the previous ones but with sigmoid activation function (output and hidden layers);SM network which are equal to the previous one but with stochastic learning process, SM network which are equal to the previous one but with stochastic learning process; Elman recurrent network with back propagation momentum learning process (ELM)and radial basis function networks (RBN)

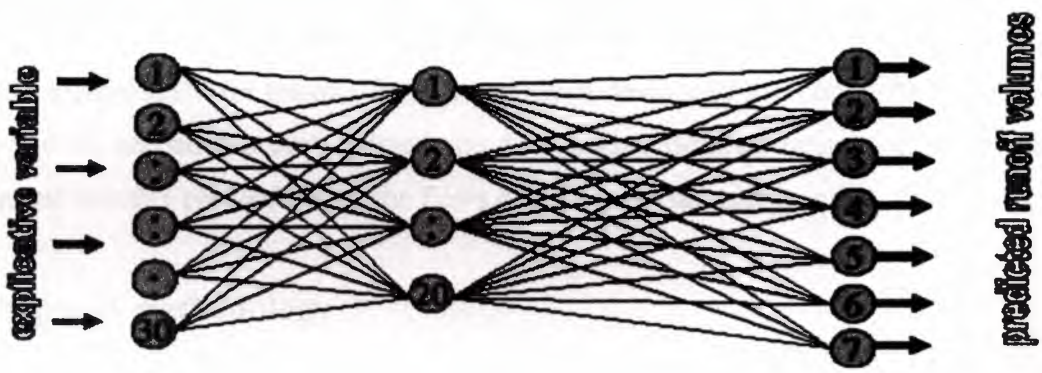


Figure 5.1. Artificial neuron net work folly connected, multi input, multi out put structure (MIMO) and (30, 20, 7)

5.4 Problem formulation

Application of ANN to stream flow predictions requires a decision regarding four main aspects: selection of the variables that best explain runoff, design of the optimal network architecture, selection of the best strategy for the learning process and selection of the system that best represents the stream flow universe. Since there are several ways in which the researcher can full the mentioned tasks, it is important to develop a systematic procedure that can produce an ANN that captures most of the predictable information present in the data and that can be safely generalized to represent realizations different to the ones present in the training episodes.

This problem proposes a method to obtain an ANN to predict monthly stream flow in mountain watersheds, subject to rainfall and snowmelt in conditions of scarce hydrologic information .The models should be able to predict stream flow one to seven months in advance during the spring and summer periods Input variables are ENSO index for Zone 3 of the Pacific, monthly temperature, precipitation and snow course information .Two different approaches were tested .The first approach developed a different ANN for each month and the second approach defined one ANN with 7 different output variables which represented monthly predictions for the flows of spring and summer months.

5.5 Proposed methodology

There is no unique and systematic methodology for the design and validation of an ANN model.

This problem presents a procedure developed using current technical literature, heuristics and experience of experts in artificial intelligence .The steps of the method are included in the block diagram presented in Figure 5.2

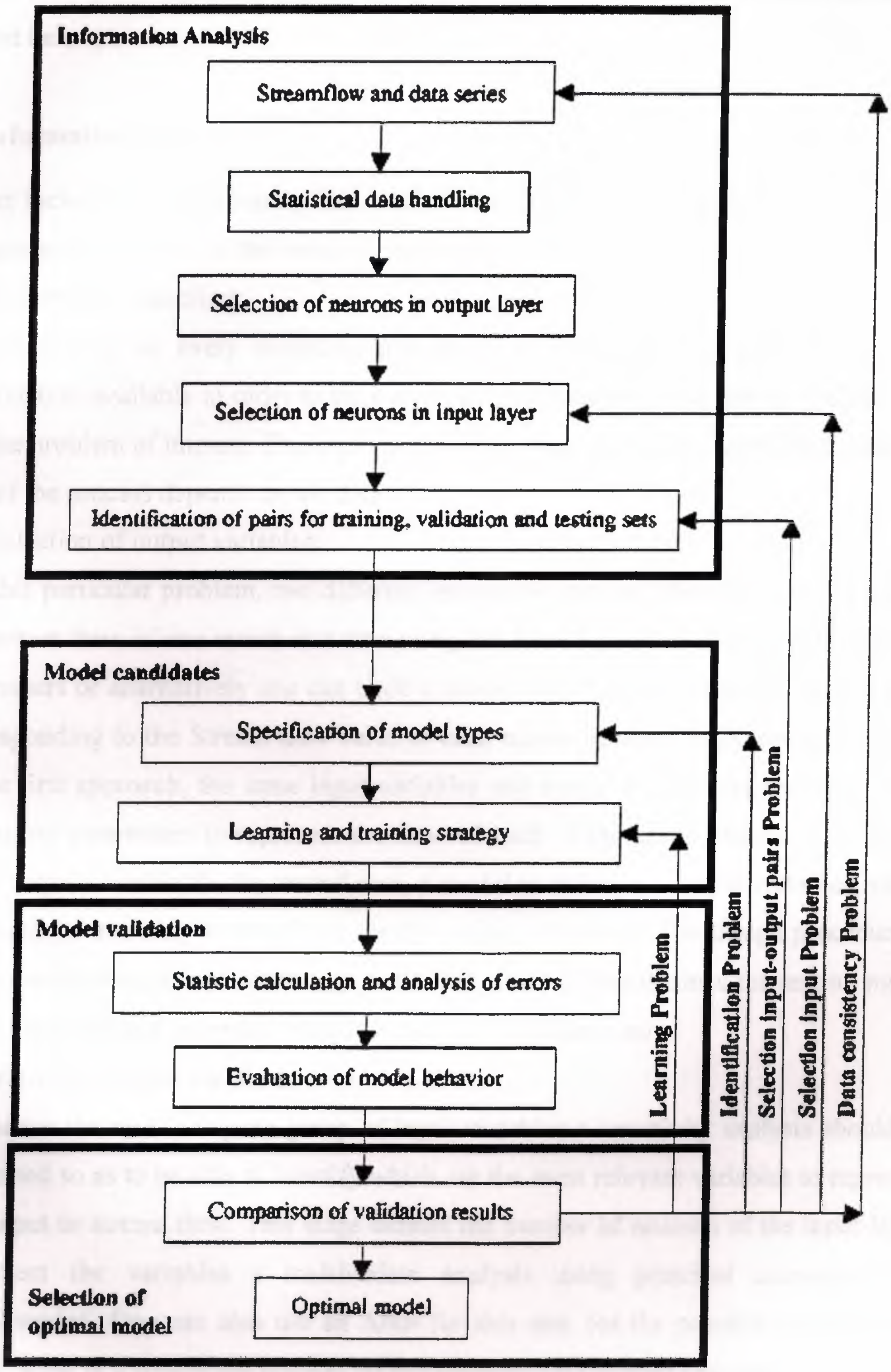

Information Analysis
Streamflow and data series
Statistical data handling
Selection of neurons in output layer
Selection of neurons in input layer
Identification of pairs for training, validation and testing sets
Model candidates
Specification of model types
Learning and training strategy
Model validation
Statistic calculation and analysis of errors
Evaluation of model behavior
Comparison of validation results
Selection of optimal model
Optimal model
Learning Problem
Identification Problem
Selection input-output pairs Problem
Selection Input Problem
Data consistency Problem

Figure 5.2. Stages in ann. model formulation

And the principal stages which are information analysis and model identification are described below:

5.5.1 Information analysis

This part includes the preliminary data analysis, the selection of the most pertinent inputs and output and the choice of the learning, testing and validation data sets

A) Information handling:

The first step in every modeling process is to collect and analyze the relevant information available in order to have accurate, representative and relevant information for the problem of interest. The importance of this stage cannot be overlooked, since the rest of the process depends on the data.

B) Selection of output variables:

For this particular problem, two different approaches can be followed .One can predict the stream flow of one month at a time using the same type of model but with different parameters or alternatively one can build a model with 7 output variables each of them corresponding to the Stream flow value of each month between Julys through January. In the first approach, the same input variables and model architecture are maintained, estimating parameters to represent the flow of each of the seven months. This model has 1 output variable. In the second case, a model to estimate 7 output variables, which represent the monthly stream flows for the period of interest .The design procedure in both cases, is the same, even though both models are different, since the second model has to represent and interpret several phenomena simultaneously.

C) Selection of input variables:

To choose the most adequate group of input variables a sensitivity analysis should be performed so as to be able to identify which are the most relevant variables to represent the output or stream flow. This stage defines the number of neurons of the input layer. To select the variables a multivariate analysis using principal components is recommended. One can also use an ANN for this step but the possible combinations tend to grow in a significant manner, so the required time to train and validate the ANN models might be non practical. Using a different type of model, as suggested, makes this stage independent of model design and leaves the possibility of performing a sensitivity analysis of the output in the validation stage of the process This sensitivity analysis might even suggest the advantage of altering the selected input variables. This step is especially critical since leaving out variables which explain a significant part of the variance present in the stream flow data should be avoided. On the other hand it is convenient to have a parsimonious model. This stage has a direct effect on the learning time of the model and the accuracy of the final result

D) Selection of learning, validation and testing sets:

Once the input and output variables are defined, it is convenient to classify the complete data set in three categories: dry, normal and wet, Depending on the flow volumes on each month, this classification was obtained by using an ANN model with the same input and output variables, excluding the hydrologic response index. The inclusion of an index of this type has been suggested in order to improve system response The selection of the input-output pairs which form the validation, training and testing sets is not random; in order to have a model with adequate predictive capability for the whole range of the data The learning and validation sets should have at least 40%of the pairs representing each of the hydrologic response categories. The testing set should include 20%of the pairs representing each hydrologic response.

5.5.2. Model identification

This step of the process includes the definition of model architecture, the learning or training process, the validation of the model and the selection of the optimal model.

A) Selection of the number of neurons in the hidden layer:

To select the number of neurons of the hidden layer the Cascade-Correlation algorithm proposed by Fahlman and Lebiere (1990) is recommended, the sets used for training, validation and testing are the same for each of the 5 different types of models mentioned previously. This algorithm combines the increment of the number of neurons in the hidden layer with the learning process, analyzing the effect on the sum of square errors in the output. This is a faster algorithm than the back propagation algorithm and is specially useful to identify the best number of neurons .For the case of the Elman network the Recurrent-Cascade-Correlation (RCC) algorithm is recommended

.Afterwards, model architectures are subject to connection pruning, using the magnitude based algorithm, in order to obtain more sparse models, Once the number of neurons in the hidden layer is defined, the models are completely specified and the selection of the best model can be approached.

B) Learning process of model candidates:

The weights necessary to represent the required output of the ANN models ,are determined through a learning process using the information represented by the input variables .The best set of weights represent the values that minimize an objective function, such as the mean square error. The following learning methods are suggested. Back propagation momentum for feed forward networks, Recurrent Cascade Correlation algorithm for Elman networks, and Radial Basis Learning algorithm for radial Basis Function. The initialization function and the way in which weights are modified during the learning process depend on the algorithm used in the learning process.

An important consideration is the criteria to terminate the learning process, in order to avoid over fitting. For feed forward networks

C) Validation and evaluation of predictive capability:

The validation process consists in the analysis of errors, defined as the difference between observed and estimated stream flow for each set of outputs (learning, validation and test sets).A useful index is the number of cycles needed for training the network. The statistics used for the objective function are the ones presented in Table5.1, and measure the goodness of .t of the model, the ability of the network to generalize or extrapolate the results outside of the range of the learning set, the presence of over fitting problems, the sensibility of the network to initial conditions and the errors due to the use of a specific combination of learning and validation sets. In particular, S4E is more sensible to maximum values, MAE measures the fit to mean values .RMSE evaluates the variance of errors independently of the sample size A high value of RMSE will usually indicate a deficiency in generalization of the network due to a bad selection of the number of hidden neurons or a weak learning process. In order to evaluate the consistency of the input-output pairs the following strategy is suggested .First the network is trained with all the pairs selected for the learning process .Then, one of the pairs is eliminated from the set and the network is trained with the rest of the pairs and the isolated pair is used as the validation set. The network is trained until a minimum sum of squared errors is obtained .This process is repeated for the rest of the pairs of the learning set. Those pairs which present convergence problems are examined in detail to detect possible inconsistencies with the rest of the pairs. If an atypical behavior is detected the possibility of eliminating this pair should be studied.

E) Selection of the optimal model:

The results obtained with the validation set for each of the selected model architectures are analyzed in order to choose the best model for the required stream flow prediction. To judge which model has the best performance, graphical and analytical comparisons can be used. One can compare time series graphs of observed and predicted monthly stream flows and dispersion diagrams of observed and calculated values. Errors or residues should be analyzed to test them for normality, independence, autocorrelation and cross correlation. Both numerical and graphical results should be considered with respect to predetermined criteria to select the best model.

Table 5.1 statistic for model cooperation

Concept	Name	Formula
Sum of square error	SSE	$\sum_{1}^{J}(OBS-CALC)$
Fourth order error	S4E	$\sum_{1}^{J}(OBS-CALC)^{2}$
Mean absolute	MAE	$\frac{\sum_{1}^{L}\lvert(OBS-CALC)\rvert}{N}$

Root Mane square error	RMSE	$\sqrt{\frac{\sum_{1}^{J}(OBS-CALC)^2}{N}}$
Efficiency coefficient	COE	$E=\left(\frac{(Variance-\mathrm{Re}\,sidues)}{(data\,\mathrm{var}\,iance)}\right)$
Mean relative	ARV	$\frac{\sum_{1}^{J}(obs-calc)^{2}}{\sum_{1}^{J}(obs-avg=obs)^2}$
Coefficient of determination	R^2	$\frac{\sum_{1}^{J}(calc-avg*obs)^{2}}{\sum_{1}^{J}(obs-avg*obs)^2}$

5.6. Application to San Juan river basin

This methodology was applied to obtain a model for stream flow prediction for San Juan River basin, Argentina, using climatologically data from Patch's meteorological station located at 1900 m of altitude. Flows were measured at Km47.3 Station which controls a 20.000 square kilometer watershed. The neural network model was used to predict monthly flows for the period from July

through January ,using meteorological information gathered during April, May and June .Input variables used were the month number ,mean temperature, relative humidity, sunshine hours, wind velocity ,snow depth, number of cloudy days ,precipitation, ENSO index and runoff volumes for the three preceding months. Output variables were the 7 monthly flows for the period July through January. The San Juan River basin is located in a mountainous region of the Andes Mountain Ranges .The area was formed at the end of the Tertiary and beginning of the Pleistocene, period in which the Andes Mountain Ranges complete their ascent and acquire their current morphology. The upper basin area extends from latitude 30 −33 ±S (Olivares Mountains) to 32 south (northern slope of the Aconcagua), and from the summits of the Tiger Mountain Range and the Western, .The upper part of the basin, which represent approximately 50% of the total area, is located in the central part of the mountain ranges of the Andes, where the altitude ranges from 3,500 m above sea level to a maximum of 7,000 m. The average elevation for the basin is 3,750 m. Precipitation on the basin is mostly snow. The liquid precipitation on the lower basin hardly reach the 100 mm per year .Although there are no rainfall stations on the upper basin, above 3,000 m, Pastilles River with 400 mm per year and Teatinos River with 300 mm per year .A study of the CFI, (1961) classifies precipitation in the Caste River basin as snow, with occasional rains; Los Pates Superior River, as glaciers and little snow; and Blanco River, with prevalence of permanent snow and with greater influence in the regime of the San Juan River. The annual average flow of the basin of the San Juan River is 2078 Mm3. With the 18 years of available data input-output pairs were prepared .Each pair consisted in 30 input variables and 7 output variables, that is, a total of 666 data elements were used for training and testing the network .From these pairs,17 were used for training and the last pair was used for testing the network .As mentioned the 30 input variables correspond to the observed values at Station Passion for the months of April, May and June, of number of the month, IOS (Index of Oscillation of the South),monthly precipitation, average temperature, relative average humidity, effective sunshine hours, average monthly wind ,maximum snow depth ,number of cloudy days and the flow volumes measured at Km 47.3 gaging station. Both stations are dependent of the Department of Hydraulics of the County of San Juan. The 7 variables correspond to the values of the monthly flow volumes measured at the Km 47.3 gaging station, for the months of July to January.

A feed-forward ANN model having an architecture 30-20-7 (30 input neurons, 20 hidden neurons and 7 output neurons) trained by means of back-propagation momentum and using the learning

Strategy suggested by Mitchell (1997) for scarce data had a good predictive behavior .Connection pruning was carried out to obtain a total of 320 connections.

All values included in the pairs were scaled to 0.7 with the purpose of homogenizing the magnitudes of the different variables, to be able to use a sigmoid function as an activation function at the network exit, and to extrapolate output values greater than the ones used in the training of the network. The network was initialized with random values between1.

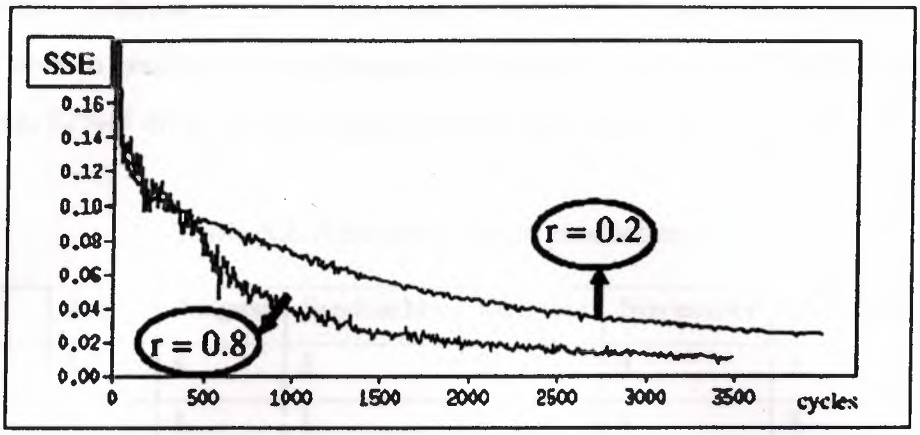


Figure 5.3. Training curves for different strategies using bock propagation momentum method

The sum of squared errors (SSE) is shown in Table 5.2 for the prediction of stream flow as a function of the number of training cycle's .Results show that the network has a good generalization performance using 4000 training cycles

Table 5.2. Sum of square errors as a function of training cycles

CYCLES	*SSE*
500	0.003772
1300	0.04307
2000	0.01719
4000	0.01482
4000 IOS	0.00900

Table 5.3 presents the absolute values of prediction errors in percent and the frequency of different error magnitudes for the 17 year period .It can be observed that errors are only in

two cases greater than 14%.Figure 4 shows the mean errors for the 7 month prediction period for two different number of training cycles .This curves indicate the ability of the neural network to generalize its performance. The graph shows that if 4000 cycles are used for Training, 82%of the prediction instances have absolute errors inferior to 5%.

Table 5.3. Absolute value of production error

year	July	August	September	October	November	December	January
1981	2	6	5	5	4	3	1
1982	1	1	1	0	0	0	0
1983	0	2	2	0	0	0	0
1984	1	2	1	0	0	0	0
1985	5	14	11	4	1	1	0
1986	4	2	4	1	0	0	0
1987	1	2	1	1	0	0	0
1988	0	0	0	1	0	1	0
1989	2	10	7	4	1	2	1
1990	9	4	2	1	0	0	0
1991	0	2	1	1	0	0	0
1992	4	3	8	5	1	1	1
1993	0	3	7	7	1	3	1
1994	6	11	8	4	0	0	1
1995	0	2	2	0	0	0	0
1996	8	6	6	9	1	4	4
1997	6	11	3	22	4	1	17
Error<5%	82%	65%	65%	71%	100%	100%	94%
Error<10%	100%	82%	94%	94%	100%	100%	94%
Error<15%	100%	100%	100%	94%	100%	100%	94%
Error<20%	100%	100%	100%	94%	100%	100%	100%

Figure 5 shows observed and predicted flow values it can be observed that the neural network represents very closely the measured flows and hence constitutes a good approach to. Flow prediction

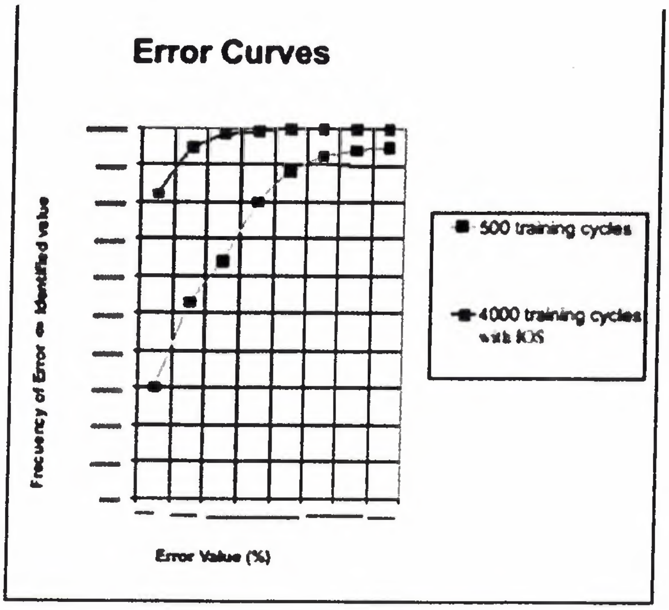


Figure 5.3. Error curves

5.7. Summary

In this chapter we have describe systematic methodology has been proposed to develop neural networks for flow prediction .An application to predict flows during the spring and summer season for the San Juan River basin shows a very good performance of the model .In this case, the use of global indicators such as the Southern Pacific temperature oscillation, has improved the predictive capacity of the model. The back propagation momentum method and the strategy suggested

By Mitchell for cases of scarce information were used The SNNS software proved to be a valuable and easy to use tool, for model identification and validation. Calculated flows show that monthly spring and summer stream Flow represented by neural network models has a better performance than alternative procedures .The proposed method has significant benefits for optimal use of water resources for irrigation and hydroelectric power generation

CONCLUSION

In studying "intelligence" we have not limited ourselves to one aspect, such as vision, problem solving, or expert systems, but we have studied the total intelligent system. This system has senses, objectives, a selection of responses from memory, a possibility to act on its environment, and finally the ability to learn from its experiences. Thus intelligence, in the sense that we defined it, is basically a stimulus - response mechanism, but with a selection of responses according to an objective. It is a way to choose an adequate response to a given situation, a response that brings the system nearer to its objective.

It is amazing how little an A.I.S., can really know about its environment. The IS builds up an internal representation of its environment to the best of its ability and in doing so, creates its own concepts, it is of the utmost importance that incoming information be checked. We discovered that while sensory information is often limited, it is (mostly) reliable. Information received from another A.I.S, however, is often incorrect, and most often unintentionally so. It must thus be carefully checked. The study of A.I.S. has also shown what their "world view" could be. Again it is seen that what is true for the A.I.S. is also true for human beings.

In chapter one, Two important questions be asked : are you concerned with thinking or behavior? Do you want to model humans, or work from an ideal standard? Computer engineering provided the artifact that makes A.I. applications possible. A.I. programs tend to be large, and they could not work without the great advances in speed and memory that the computer industry has provided.

Does humanity's new knowledge about artificial intelligence systems affect its future? The knowledge about A.I.S. is currently rudimentary and it looks like many years will be required to understand intelligent systems fully. Perhaps in a hundred years, we will still be increasing our knowledge on intelligent systems.

In chapter two, "Expert Systems" are a part of general category of computer applications known as Artificial Intelligence to design an expert system, one needs a knowledge engineer, an individual who studies how human experts make decisions and translates the rules into terms that a computer can understand many "real time" expert systems are 'soft' real-time systems, in that they claim to be fast. A 'hard' real-time system would have features that guarantee a response within a fixed amount of real-time (e.g. bounded computation, not just a fast match-recognizes –act cycle).

In chapter three, "fuzzy logic" logic of fuzzy is working by considering a small example about it as you have read, and we have considered also important method to make it easy to understand fuzzy logic, And we have entered to fuzzy modeling by consider the way we have to use in modeling and the steps it have to be started in our modeling, and also we consider same problem fuzzy logic control and concept action to solved some like of this kinds. And in end it become, the way of how we can develop fuzzy logic

In chapter four, "Neural Networks" are based on the parallel architecture of animal brains. Neural networks will be the future in computing. Real brains are orders of magnitude more complex than any other artificial neural network o far considered. A great deal of research is going on in neural networks worldwide

In chapter five, "Applications" every application of these types of artificial intelligence do well in its own job, so applications that are mentioned in this chapters are not the all applications in the life there are many different applications. These applications differ from each other, so we can make a comparison between them.

This project aimed at investigating the A.I. system, the fins Tony has been and development of A.I. system investigating .A reel life application on A.I. was presented.

The area of artificial intelligence is becoming more important in both undergraduate and graduate curricula in computer science and engineering ,this will provide the fundamental conceptual necessary to confront the rapidly developing of the world .

REFERENCES

[1] http://tqd.advanced.org/2705/
history.html

[2] www.aaai.org/aitopics/html/history.html

[3] www.aaai.org/aitopics/html/history.html

[4] www.aaai.org/aitopics/html/history.html

[5] Daniel Creviert's book," the history of the artificial intelligence, an engineering approach"

[6] Daniel Creviert's book," the history of the artificial intelligence, an engineering approach"

[7] Rosenblatt, F.Rosenblatt, "The perceptron: A probabillstic model for information storage and organization in the brain", Psychological Review, 65:386-408, 1958

[8] Haykin, S. and Li, L., "Nonlinear adaptive prediction of no nstationary signals". IEEE Transactions on Signal Processing, 43(2):526-535, 1995.

[9] Kashman A., Neural Networks: "Lecture Notes of COM420", Near East University, Nicosia, 2002.

[10] B.S. Hoffheins, Using Sensor Arrays and Pattern Recognition to Identify Organic Compounds. MS-Thesis, the University of Tennessee, Knoxville, TN, 1989

[11] Parker, D.B., "Learning-Logic", MIT Center for Computational Research in Economic and Management Science, Cambridge, MA, 1985

[12] J.F. Baldwin, "Fuzzy logic and fuzzy reasoning," in Fuzzy Reasoning and Its Applications, E.H. Mamdani and B.R. Gaines (eds.), London: Academic Press, 1981.

[13] An introduction to neural computing. Aleksander, I. and Morton, H. 2nd edition

[14] An introduction to neural computing. Aleksander, I. and Morton, H. 2nd edition

[15] Learning internal representations by error propagation by Rumelhart, Hinton and Williams (1986).

[16] Industrial Applications of Neural Networks (research reports Esprit, I.F.Croall, J.P.Mason)

[17] Klimasauskas, CC. (1989). The 1989 Neuron Computing Bibliography.

[18] Neural Networks, Eric Davalo and Patrick Naim

[19] Alkon, D.L 1989, Memory Storage and Neural Systems, Scientific American, July, 42-50

[20] A brahart R.J., S ee L., K neale P.E. (1998)New Tools for Neuron hydrologists :using network pruning and model breeding