NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

ARTIFICIAL INTELLIGENCE SYSTEMS

Graduation Project
COM-400

Student: Nashaat Ghuneim

Supervisor: Assoc. Prof. Dr Adnan Khashman

Nicosia - 2002

AKNOWLEDGEMENT

First of all I am so thankful to most gracious Allah, the Almighty, who enabled me to complete my project.

I would like to thank my parents, sisters, brothers and all my relatives for their encouragement and lawsuits till got what I repairing and standing at.

I should not forget to thank my intellectual and conscientious **Assoc. Prof. Dr Adnan Khashman** for inspiring and endurance me to complete this project.

With homologizing, I would like to thank my dear university (N.E.U.) that never forgot its big unmistakable and contribution in my development and composing my self.

I want to say thanks to all of my friends who helped me in finalizing my project and especially to all I have living with, for their standing beside me. Really they were my shadow I twigged in complete my project.

My brothers Ihsan and Muzher Ghuneim,Yousef Ghneim,Yousef abu Khurj and Mohammed Sharaf

ABSTRACT

This project explores the theoretical and particular underpinning of Artificial Intelligence (A.I.) and provides an introductory-level on A.I. technology.

The reader of this project will come away with an appreciation for the basic concepts of A.I., and also with an idea of what can and cannot be done with today's technology, at what cost and using what techniques.

This project reports on the area of Artificial Intelligence and how it become more important in both undergraduate and graduate in computer science and engineering. Hopefully, this will provide the fundamental conceptual necessary to confront the rapidly developing of the world.

This project include a general background and history about the artificial intelligence and some early examples, it concentrates on artificial intelligence system and how they work, some applications areas is also included in this project and the theme of A.I..

In this project several model and techniques of A.I.S. available; like Neural Networks, Expert Systems, Genetic Algorithms and how they work, some advantages and disadvantages, there applications areas and a comparison between these applications in the real life.

TABLE OF CONTENTS

INTODUCTION

This project concerns the foundations of a new generation of computing technology and capability, most commonly referred to as Artificial Intelligence (Al).

Programs providing capabilities like English-language communication expert reasoning and problem solving, and even master level chess skill are having a profound effect on how people use computers and what they use them for.

The objectives of this thesis can be summarized as:

- Investigation a general overview, history and the development of Artificial Intelligence.
- Description of various Artificial intelligence fields (Neural Networks, Expert System, genetic algorithms and applications on each one of these fields).
- Comparison of the various Artificial Intelligence fields.
- Investigation of different types of life applications for artificial intelligent systems.

Thesis structure

In chapter one, a general background and history of "Artificial Intelligence" will present, objectives and some early example of Artificial intelligence system.

In chapter two, different types of applications areas of artificial intelligence will be present and simple description about the modern applications. The themes, the concepts, the process and the mechanism of A.I. are also mentioned in this chapter.

In chapter three, "Expert Systems" describes a general history, advantages and disadvantages of expert systems. What the system does, examples of expert system, development and developing of an expert system.

In chapter four, "Neural Networks" introduction to N.N., general definitions, how the artificial neuron work supervised and unsupervised training. The benefits of neural networks are also mentioned, different applications of neural network and the difference between neural networks, traditional computing and the expert systems.

In chapter five, "Genetic Algorithms" describes a general history, some definitions and how the genetic algorithms work in real life. Representation of artificial intelligence with genetic algorithms, how they work together and different applications of genetic algorithms.

In chapter six, represent different applications of each chapter and comparison between thee applications.

CHAPTER ONE

ARTIFICIAL INTELLIGENCE

1.1. Overview

Researchers in the science of "Artificial Intelligence" have investigated many areas of the mind such as pattern matching, vision, and theorem proving. However, all of these are only parts of the human mind. An intelligent system could include all of these parts, but it still would not be complete, and could not function, unless it also had senses, 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. Background and History of Artificial Intelligence System

In order to classify machines, as "thinking" it is necessary to define intelligence. To what degree does intelligence consist of, for example, solving complex problems, or making, or making generalizations and relationships? In addition, what about the perception and comprehension? Research into the areas of learning, of language, and of sensory perception has aided scientists in building intelligent machines. One of the most challenging approaches facing experts is building system that mimics the behavior of the human brain, made up of billions of neurons, and arguably the most complex matter in the universe. Perhaps the best way to gauge the intelligence of a machine is British computer scientist Alan_Turing's_test. He stated that a computer would deserve to be called intelligent if it could deceive a human into believing that it was human.

Artificial Intelligence has come a long way its early roots, driven by dedicated researchers. The beginnings of AI reach back before electronics, to philosophers and mathematicians such as Boole_and others theorizing on principles that ere used as the foundation of AI logic. AI

The technology was finally available, or so it seemed, to simulate intelligent behavior. Over the next four decades, despite many stumbling blocks, AI has grown a dozen researchers, to thousands of engineers and specialists; and from programs capable of playing checkers, to systems designed to diagnose disease.

A.I. has always been on the pioneering end of computer science. Advanced –level computer languages, as well as computer interfaces and word –processors over their existence to the research into artificial intelligence. The theory and insights brought about by AI research will set the trend in the future of computing. The products available today are only bits and pieces of what are soon to follow, but they are a movement towards the future of artificial intelligence. The advancements in the quest for artificial intelligence have, and will continue to affect our jobs, our education, and our lives.

In 1950 Alan Turing, the mathematician, proposed an elegant test (the Turing Test) for machine intelligence: if a machine can carry out a lengthy conversation (via a keyboard, for example) with a human, and the human is unable to tell whether the conversation is with a machine or other test is that humans may make poor judges. Simple tricks can be used to fool unsophisticated observer for a short time but it is likely that to be successful over a sustained interrogation a number of significant artificial intelligence challenges will need to be overcome, including codification of common sense, and machine understanding of natural language.

The victory of IBM’s deep blue over chess grandmaster Kasparov in May 1997 was a landmark in machine intelligence. Nevertheless, Deep Blue had several advantages, including access to all Kasparov’s previous games and a team of experts that could make overnight modifications to its program. Intelligence involves knowledge, pattern recognition, deductive reasoning, and learning from experience. Different approaches to artificial intelligence place varying degrees of emphasis on these aspects.

Humankind has given itself the scientific name bombo sapiens _man the wise because our mental capacities are so important to our everyday lives and our sense of self. The field of artificial intelligence, or AI, attempts to understand intelligent entities. Thus, one reason to study it is to learn more about ourselves. But unlike philosophy and psychology, which are also concerned with intelligence, AI strives to build intelligent entities as well as understand them. Another reason to study AI is that these constructed intelligent entities are interesting and useful in their own right. AI has produced many significant and impressive products even at this early stage in its development. Although no one can predict the future in detail, it is clear that computer with human-level intelligence would have a huge impact in our day lives and on the future course of civilization.

The study of intelligence is also one of the oldest disciplines. For over 2000 years philosopher have tried to understand how seeing, learning remembering, and reasoning could, or should, be done. The advent of usable computers in the early of 1950sturned the learned armchair speculation concerning these mental faculties into a real experimental and theoretical discipline. Many felt that the new Electronic Super Brains had unlimited potential for intelligence. But as well as a providing a vehicle for creating artificially intelligent entities, the computer provides a tool for testing theories for intelligence, and many theories failed to withstand the test. AI has turned out to be more difficult than many at first imagined and modern ideas are much richer, more subtle, and more interesting as a result.

AI currently encompasses a huge variety of subfields, from general purpose areas such as perception and logical reasoning, to specific tasks such as playing chess, proving mathematical theorems, writing poetry, and diagnosing disease. Often scientists in other fields move gradually into artificial intelligence, where they find the tools to systematize and automate the intellectual tasks on which they have been working all their lives. Similarly, workers in AI can choose to apply their methods to any area human intellectual this sense, it is truly a universal field.

Clearly there are two fairly distinct positions. The first is concerned with intelligence, containing elements of creativity, the ability to be unpredictable and with no reliance on interfacing with the outside world. The second position is concerned with the ability that machines/computers have, and how they can be, employed in way which we regard as intelligent behavior, often in the form of a conditioned response. Looking at this second position as being the present standing for Artificial Intelligence, this is very much in the flavor of Minsky. So Artificial Intelligence as described by Minsky does not involves artificially achieving intelligence, i.e. it does not mean what it says, but rather it involves doing the best the available machinery.

The general framework for intelligence can be considered two separate approaches, automated reasoning and pattern understanding. The first of these involves problem solving, such as by means of an expert system and playing; whereas the second involves image processing, pattern recognition and sensor processing. In both of these approaches a constrained real world environment is employed by the computer/machine such that a solution/pattern can be found by using available stored knowledge, generally involving a search through a number of possible solution paths.

1.3. History With Respect to Computer Engineering

For artificial intelligence to succeed, we need two things: intelligence and an artifact. The computer has been unanimously acclaimed as the artifact with the best chance of demonstrating intelligence. Scientists in three countries embattled in World War II invented the modern digital electronic computer independently and almost simultaneously. The first operational modern computer was the Heath Robinson, built in 1940 by Alan Turing's team for the single purpose of deciphering German messages. When the Germans switched to a more sophisticated code, the electromechanical relays in the Robinson proved to be too slow, and a new machine called the Colossus machines were in everyday use.

The first operational *programmable* computer was the z-3, the invention of Koran Zuse in Germany in 1941. Zuse invented floating-point numbers for the z-3, and went to on in 1954 to develop Plankalul, the first high- level programming language. Although Zuse received some support from the Third Reich to apply his machine to aircraft design, the military hierarchy did not attach as much importance to computing as did its counterpart in Britain.

In the United States, the first *electronic* computer, John Atanasoff and his graduate student Clifford Berry assembled the ABC, between 1940 and 1942 at Lowa State University. The project received little support was abandoned after Atanasoff became involved in military the Mark I, II, and III computers were developed at Harvard by a team under Howard Aiken; and the ENIAC was developed at the university of Pennsylvania by a team including John Mauchly and John Eckert. ENCIAC was the first general- purpose, electronic, digital computer. One of its first applications was computing artillery-firing tables. A successor, the ADVAC, followed John Von Neumann's suggestion to use a stored program, so that technicians would not have to scurry about changing patch cords to run a new program.

Each generation of computer hardware has brought in an increase in speed and capacity, and a decrease in price. Computer engineering has been remarkably successful, regularly doubling performance every two years, with no immediate end is sight of this rate of increase. Massively parallel machines promise to add several more zeros to the overall throughput achievable.

Of course, there are calculating devices before the electronic computer. The abacus is roughly 7000 years old. In the mid-17th century, Blaise Pascal built mechanical adding and subtracting machine called the Pascaline. Leibimz improved on this in 1964, building a mechanical device that multiplied by doing repeated addition. Progress stalled for over a century until Charles Babbage (1792-1871) dreamed that logarithm tables could be computed by machine.

AI also owes a debt to the software side of computer science, which has supplied the operating systems, programming languages, and tools needed to write modern programs (and papers about them). But this is one area where the debt has been repaid: work in AI has pioneered sharing, interactive interpreters, the linked list data type, automatic storage management, and some of the key concepts of object-oriented programming and integrated program development environments with graphical user interfaces.

1.4. What is Artificial Intelligence system?

Intelligence and Artificial Intelligent of fundamental concern in the application of artificial intelligent is the question 'What is artificial intelligence?', and providing a straightforward, simple definition has led to much philosophical discussion. The term 'Artificial' is perhaps simple enough to understand, this meaning 'contrived, synthetic, man-made', but what Intelligence? Despite the fact the although we are intelligent, we do not really know that intelligence has been in existence for approximately 40 years and provides us with a working, powerful approach for tackling problems.

1.4.1. General Definitions

It appears that the origins of AI may be traced back to conference at Dartmouth College in the summer of 195. Perhaps the broadest definition is that:

- AI is the field of study that seeks to explain and emulate intelligent behavior in the terms of computational processes.

[From an engineer viewpoint that AI is about]

- Generating representation and procedures that automatically solve problems therefore solved by humans.

1.4.2. 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 the study of intelligence in machines and through computers, in people.
- The exciting new effort to make computers think, machines with mind, in the full and literal sense (Haugeland, 1985).
- The automation of activities that we associate with human thinking, activities such as decision-making, problem solving learning (Bellman, 1978).
- The study of mental faculties through the use of computational models (Charniak and McDermott, 1985).

- A field of study that seeks to explain and emulate intelligent behavior in terms of computational processes (Schalkoff, 1990).
- The study of how to make computers do thing at which, at the moment people are better (Rich and Knight, 1991).
- The branch of computer science that is concerned with the automation of intelligent behavior (Luger and Subblefield, 1993).

1.5. 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.

1.6. More History about Artificial Intelligence System

In 1943, mathematicians *Warren McCulloch* and *Walter Pitts* showed how it was possible for a neural network to compute. Six years later (Donald Hebb) showed how a neural net could learn. If there is a "core" to AI, today it is probably the connectionist school neural nets in that sense, *McCulloch, Pitts* and *Hebb* can be considered founding fathers of A.I.

In 1950 Alan Turing, the mathematician, proposed an elegant test (the tuning test) for machine intelligence: if a machine can carry out a lengthy conversation (via a keyboard, for example) with a human, and the human is unable to tell whether the conversation is with a machine or another human, the machines is said to be exhibiting intelligence. One weakness of the test is that humans may make poor judges. Simple tricks can be used to fool the unsophisticated observer for a short time but it's likely that to be successful over a sustained interrogation a number of significant artificial intelligence challenges will need to overcome, including codification of common sense, and machine understanding of natural language.

But connectionism was not always central to AI. In the early decades, most research focused on symbolic, rule-based reasoning, also known as expert systems. This article draws mainly upon *Daniel Crevier* excellent 1993 book, AI: The Tumultuous History of the Search for Art facial Intelligence for a review of AL up to the maturity of expert systems into real-world problem solvers.

Herbert Simon was a political scientist and expert in bureaucratic organization and to some extent economics. He proposed the theory of "satisficing" that we in fact make decisions without bothering to go to all the trouble of gathering information about all the options which led to the notion of heuristics or "rules of thumb" developed by *George Polya* in 1945. Heuristics have since been shown to be an essential element in both human and artificial intelligence.

Newell and *Simon* followed up their success with Logic Theorist in 1957 by creating a program which unlike Logic Theorist was not preprogrammed for a specific task (in Logic Theorist's case, the task was proving math theorems). The new program was called General Problems Solver (GPS) for that reason. GPS made more use of feedback to refine a solution by an iteration process. GPS tell us to "learned to solve various puzzles, performed symbolic integration and broke secret codes."

A.I. as a discipline was founded that same year 1956 at what has become known as the *Dartmouth conference*. The general belief at that time was that intelligence could be simulated in a machine.

In the late 1950s/early 1960s, *Robert K. Lindsay* created "Sad Sam," a program that could read English sentences and draw conclusions from them, but only in a very limited domain. Given the sentences: "Jim is John's brother" and "Jim's mother is Mary," Sad Sam was smart enough to understand that Mary must therefore be John's Ma. This was a step in the right direction. As humans, "to understand is to be able to relate a new piece of information to other facts we already know.

1.7. Early Examples of Artificial Intelligence System

Here is an example of a session with SIR quoted by *Crevier* (SIR's responses are in capitals):

1. Every boy is a person.

I UNDERSTAND

2. A finger is part of a hand.

I UNDERSTAND

3. Each person has two hands.

THE ABOVE SENTENCE IS AMBIGUOUS. PLEASE RE-PHRASE IT.

4. There are two hands on each person.

I UNDERSTAND

5. How many fingers does John have?

THE ABOVE SENTENCE IS AMBIGUOUS, BUT I ASSUME (HAS) MEANS

(HAS AS PARTS). I DON'T KNOW WHETHER FINGER IS PART OF JOHN. 6. John is a boy.

I UNDERSTAND

7. How many fingers does John have?

THE ABOVE SENTENCE IS AMBIGUOUS, BUT I ASSUMED (HAS) MEANS (HAS AS PARTS), HOW MANY FINGERS PER HAND?

8. Every hand has five fingers.

I UNDERSTAND

9. How many fingers does John have?

THE ANSWER IS 10.

Work at MIT in the mid-to-late 1 960s focused on getting computers to manipulate blocks, which meant they had to understand three-dimensional geometry and certain aspects of physics. And they had to be able to see. The problem of how to make a computer not only see, through video cameras, but more importantly how to make it makes sense of what it sees, was tackled by a variety of researchers at MIT including *Larry Roberts, Gerald Sussman, Ado/Jo Guzman, Max C/owes* and *David Huffman, David Waltz, Patrick Winston,* and *Berthold Horn.* The end result of their efforts was micro-blocks world," where a robot was able to see the set of blocks on the table and move and stack them.

Numerous refinements to the control programs were made over the years. Each tiny improvement took a lot of effort. A program called STRIPS took the lead over GPS for a while, then along came *Terry Weinograd's* SHRDLU (a nonsense name -- it has no meaning.)

SHRDLU was more than an incremental advance -- it was a considerable advance. It let humans interrogate the robot in a block world. Here is an example, again taken from Crevier (SHRDLU's responses in capitals):

1. Pick up a red block.

OKAY

2. Find a block, which is taller than the one you are holding and put it into the box.

BY "IT," I ASSUME YOU MEAN THE BLOCK WHICH IS TALLER THAN THE ONE I AM HOLDING. OKAY

3. What does the box contain?

THE BLUE PYRAMID AND THE BLUE BLOCK

4. What is the pyramid supported by?

THE BOX

5. How many blocks are not in the box?

FOUR OF THEM

6. Is at least one of them narrower than the one, which I told you to pick up?

YES, THE RED CUBE

7. Is it supported?

YES, BY THE TABLE

8. Can the table pick up blocks?

NO

9. The Blue pyramid is mine

I UNDERSTAND

10. I own all blocks, which are not red, but I don't own anything, which supports a pyramid

UNDERSTAND

11. Do I own the box?

NO

12. Do I own anything in the box?

YES, TWO THINGS: THE BLUE BOX AND THE BLUE PYRAMID

13. Will you please stack up both of the red blocks and either a green cube or pyramid?

OKAY

14. Is there a large block behind a pyramid?

YES, THREE OF THEM: A LARGE RED ONE, A LARGE GREEN CUBE AND THE BLUE ONE

15. Put a small one onto the green cube which supports a pyramid

OKAY

16. Put the littlest pyramid on top of it

OKAY

17. Had you touched any pyramid before you put the green one on the little cube?

YES, THE GREEN ONE

18. When did you pick it up?

WHILE I WAS STACKING UP THE RED CUBE, A LARGE RED BLOCK AND A LARGE GREEN CUBE

19. Why?

TO GET RID OF IT

20. Why did you do that?

TO CLEAR OFF THE RED CUBE

21. Why did you clear off that cube?

TO PUT IT ON LARGE GREEN CUBE

22. Why did you do that?

TO STACK UP THE RED CUBE, A LARGE RED BLOCK AND A LARGE GREEN CUBE

23. Why did you do that?

BECAUSE YOU ASKED ME TO

As so far, the history of A.I. has had cycles of success, misplaced optimism, and resulting cutbacks in enthusiasm and funding. There have also been cycles of introducing new creative approaches and systematically refining the best ones. I have known explained why A.I. is exciting.

1.8. Main Objectives of Artificial Intelligence System

In natural I.S., the random perils of their environment have resulted in the determination of their main objective as being the survival of the species, as was shown by Charles Darwin.

In artificial ISs, for instance in robots, the main objective is determined by the writer of the IS's software. In some of his books, Isaac Asimov suggested and tested a triad of main objectives which he called the *"The Three Laws of Robotics"*. They are:

1. *A robot may not injure a human being, or, through inaction, allow a human being to come to harm.*
2. *A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.*
3. *A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.*

1.9. Summary

This chapter defines A.I. and establishes the cultural background against which it has developed. Some of important points are as follows:

- Different people think of AI differently. Two important questions are: 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 AI applications possible. AI programs tend to be large, and they could not work without the great advances in speed and memory that the computer industry has provided.
- The history of AI has had cycles of success, misplaced optimism, and resulting cutbacks in enthusiasm and funding. They have also been cycles of introducing new creative approaches and systematically refining the best ones.
- Recent progress in understanding the theoretical basis for intelligence has gone hand in hand with improvements in the capabilities of real system.
- Artificial intelligence was coined in 1956 by john McCarthy at the Massachusetts Institute of Technology (MIT).

CHAPTER TWO

AREAS OF APPLICATIONS

2.1. Overview

Artificial Intelligence, or A.I. for short, looms large in the present and future of both computer science and industry. The application merging from university research laboratories are making their way to commercial products with increasing speed. Programs providing capabilities like English-language communication expert reasoning and problem solving, and even master level chess skill are having a profound effect on how people use computers and for what they use them.

2.2. Theory of Artificial Intelligence System

This theory is the outcome of having built several artificial intelligent systems (IS) on a computer. Studying artificial I.S.s has the advantage over studying human ones that we can readily observe all their internal processes: we can observe the creation and use of concepts and of response rules. Thus, this theory talks about the creation of concepts, the elaboration of the present situation, the elaboration, storage and retrieval of response rules, the selection of an adequate response rule, and finally, the execution of the response part of the selected response rule.

We can observe that a surprising number of the brain functions of the human IS are quite similar to those of an artificial IS. From this, we would probably want to conclude that this is obvious, since most artificial ISs are modeled on the natural, the human one. However, this is not quite true. When we review the artificial intelligence literature we can observe that a wide variety of approaches have been utilized in the functional design of artificial ISs. However, the author and others have noted that most of these other approaches do not work well, even if at first sight they seem to be quite reasonable. Further examination shows that many of those that have worked show these amazing similarities to how we currently believe the human mind functions.

2.3. Branches of Artificial Intelligence

We will make what I believe to be reasonable assumption: that true A.I.is at least theoretically possible. After all, you would not have read this far you did not have some belief in possibility that a computer can mimic the reasoning of human experts.

2.3.1. What are the branches of Artificial Intelligence?

Here's a list, but some branches are surely missing, because no-one has identified them yet. Some of these may be regarded as concepts or topics rather than full branches.

1. Logical A.I.

What a program knows about the world in general the facts of the specific situation in which it must act, and its goals are all represented by sentences of some mathematical logical language. The program decides what to do by inferring that certain actions are appropriate for achieving its goals. The first article proposing this was [McC59]. [McC89] is a more recent summary. [McC96b] lists some of the concepts involved in logical A.I. [Sha97] is an important text.

2. Search

AI programs often examine large numbers of possibilities, e.g. moves in a chess game or inferences by a theorem proving program. Discoveries are continually made about how to do this more efficiently in various domains.

3. Pattern recognition

When a program makes observations of some kind, it is often programmed to compare what it sees with a pattern. For example, a vision program may try to match a pattern of eyes and a nose in a scene in order to find a face. More complex patterns, e.g. in a natural language text, in a chess position, or in the history of some event are also studied. These more complex patterns require quite different methods than do the simple patterns that have been studied the most.

4. Inference

From some facts, others can be inferred. Mathematical logical deduction is adequate for some purposes, but new methods of *non-monotonic* inference have been added to logic since the 1970s. The simplest kind of non-monotonic reasoning is default reasoning in which a conclusion is to be inferred by default, but the conclusion can be withdrawn if there is evidence to the contrary. For example, when we hear of a bird, we man infer that it can fly, but this conclusion can be reversed when we hear that it is a penguin. It is the possibility that a conclusion may have to be withdrawn that constitutes the non-monotonic character of the reasoning.

5. Common Sense Knowledge and Reasoning

This is the area in which AI is farthest from human-level, in spite of the fact that it has been an active research area since the 1950s. While there has been considerable progress, e.g. in developing systems of *non-monotonic reasoning* and theories of action, yet more new ideas are needed. The Cyc system contains a large but spotty collection of common sense facts.

6. Learning from Experience

Programs do that. The approaches to AI based on *connectionism* and *neural nets* specialize in that. There is also learning of laws expressed in logic. [Mit97] is a comprehensive undergraduate text on machine learning. Programs can only learn what facts or behaviors their formalisms can represent, and unfortunately learning systems are almost all based on very limited abilities to represent information.

7. Planning

Planning programs start with general facts about the world facts about the particular situation and a statement of a goal. From these, they generate a strategy for achieving the goal. In the most common cases, the strategy is just a sequence of actions.

8. Ontology

Ontology is the study of the kinds of things that exist. In AI, the programs and sentences deal with various kinds of objects, and we study what these kinds are and what their basic properties are. Emphasis on ontology begins in the 1990s.

9. Heuristics

A heuristic is a way of trying to discover something or an idea imbedded in a program. The term is used variously in AI. *Heuristic functions* are used in some approaches to search to measure how far a node in a search tree seems to be from a goal. *Heuristic predicates* that compare two nodes in a search tree to see if one is better than the other, i.e. constitutes an advance toward the goal, may be more useful. [My opinion].

2.3.2. Applications of Artificial Intelligence system

There are many applications on A.I. that can be summarized by the following applications in the real life.

Game Playing

You can buy machines that can play master level chess for a few hundred dollars. There is some AI in them, but they play well against people mainly through brute force computation-looking at hundreds of thousands of positions. To beat a world champion by brute force and known reliable heuristics requires being able to look at 200 million positions per second.

Speech Recognition

In the 1990s, computer speech recognition reached a practical level for limited purposes. Thus United Airlines has replaced its keyboard tree for flight information by a system using speech recognition of flight numbers and city names. It is quite convenient. On the the other hand, while it is possible to instruct some computers using speech, most users have gone back to the keyboard and the mouse as still more convenient.

Understanding Natural Language

Just getting a sequence of words into a computer is not enough. Parsing sentences is not enough either. The computer has to be provided with an understanding of the domain the text is about, and this is presently possible only for very limited domains.

Computer Vision

The world is composed of three-dimensional objects, but the inputs to the human eye and computers' TV cameras are two dimensional. Some useful programs can work solely in two dimensions, but full computer vision requires partial three-dimensional information that is not just a set of two-dimensional views. At present there are only limited ways of representing three-dimensional information directly, and they are not as good as what humans evidently use.

Heuristic Classification

One of the most feasible kinds of expert system given the present knowledge of AI is to put some information in one of a fixed set of categories using several sources of information. An example is advising whether to accept a proposed credit card purchase. Information is available about the owner of the credit card, his record of payment and also about the item he is buying and about the establishment from which he is buying it (e.g., about whether there have been previous credit card frauds at this establishment).

2.4. Mechanism of an Artificial Intelligence System

In artificial ISs, the brain first attempts to make a list of the response rules that are applicable to the present concrete situation. ("Applicable" here means that the response rule has some concepts in its situation part that also occur in the present situation.) If it does not find any response rules, it then attempts to make a list of those response rules that are applicable to the present situation when it expresses that situation with (total) concepts, meaning concepts that have concepts of the situation in its link to part concepts. If it still doesn't find any response rules, it attempts to make a list of those response rules that are applicable to the present situation when it expresses that situation with (abstract) concepts, meaning concepts that have concepts of the situation in its link to concrete concepts.

Once it has finished finding and building an appropriate response rule list, the IS continues and as expected selects a single rule which it will attempt to use. However, the process that it uses to do this, is, to us natural ISs, a rather curios one. Priding ourselves on our use of logic, we would naturally assume that an artificial IS would always (logically) choose the response rule that seems best for the given situation. However, that is not what a good' artificial IS does.

2.5. Process of an Artificial Intelligence System

As an illustration, lets take a game of chess and assume that the IS has previously learned that it is good to take a pawn with a bishop. That is, it created the corresponding response rule and gave it a high evaluation. Let us also assume that it had also previously taken a bishop with its own bishop, but that its bishop was then promptly taken in turn. As expected, with this experience the IS created a corresponding response rule and gave it a lower evaluation. Now, suppose the IS always makes the best move known to it. How would it function?

The answer is easy: Every time that it could take a pawn with its bishop, it would do so. This is because, as far as it knows, this is the best move. Nevertheless, we humans--or at least those of us who have some accumulated knowledge of chess--know that in many situations that this is not the best move. If instead of taking a pawn it can take an unprotected bishop, it would (almost always :-) be far better to take the bishop instead of the pawn. While this bishop-taking (present) situation may be similar to the previous bishop-taking situation, it is not the same. The difference, protection, is very important. If the artificial IS wants to make the best moves that it can, and then it has to be enabled and allowed to learn this!

2.5.1. Process of a "Good" Artificial Intelligence System

Thus, a well designed artificial IS does not simply "choose the move it considers best." Instead, it enables and allows learning by randomly choosing any move from a list of applicable moves that was previously constructed. Unfortunately, when it selects responses in this way, it will often choose a bad move. That is, a move that it considers bad and that really is bad. Thus, while this new process breaks the previous learning trap (and avoids falling into the trap of predictability), it also creates a new problem.

2.5.2. Process of a "Better" Artificial Intelligence System

Luckily, researchers have found that there is a way to get around this problem. In this improved process, the IS still randomly chooses any move from the, previously mentioned, list of applicable moves. However, it does not choose moves with an equal frequency. That is, it chooses those it considers as "better" more often, and those that have "less value", less often. Stating this in a more specific and mathematical language: How often each choice is chosen is linked, in direct proportion, to the value assigned to the particular move. In this way the artificial IS continues to learn steadily, but also still selects reasonably good moves most of the time.

2.6. Central Themes in Artificial Intelligence System

A number of themes recur in the study of AI systems. Fundamental to AI system development is the concept of:

- Knowledge representation, structure, "meaning", and acquisition.

Other basic and related themes are:

- Inference/control (manipulation) strategies.
- Ability to learn/adapt (from experience, examples, or a "teacher").
- Representation of uncertainty and incomplete reasoning.
- Search and matching techniques.
- No monotonic reasoning (retracting conclusions based on newly revised information or beliefs).
- Empiricism ("generates and test ").
- Problem decomposition, reducing overall goals to sub goals.
- Problem dynamics (changes with time).
- Type of reasoning (e.g. deduction induction and common sense).
- Satisfaction with "good" versus optimal solutions; also, question concerning the existence of a solution.
- Relevant programming/representation languages for implementation and associated architectures.

2.7. The Field of Artificial Intelligence System

As far as we know, human intelligence is the pinnacle of progress to date in the still-unfolding story of the cosmos. It is the most complex entity we know of, because it is the latest development in a process of complexity heaped upon, or multiplied by, complexity.

First, I would like to look at the big picture of the fundamental scientific progress leading to intelligence, and then I focus in on the specific modern disciplines and technologies that our intelligence is using to create artificial intelligence. Finally, I map the structure of the field of A.I onto the structure of this project.

The laws of quantum physics turned part of whose energy into a handful of quarks. The laws of chemistry ordered the atoms of various sorts to combine in various ways to form molecules of various sorts. Mechanical laws (later described by Newton) and space-time relativity laws ordered the molecules to shuffle around to form the stuff of space, of galactic swirls, stars, and planets of various sorts.

These laws themselves combined to order the formation of a pre-biotic “soup” on Earth. New and more complex laws were devised to govern the biological and evolutionary development of organic life of various sorts. Finally, laws we can barely discern, let alone fathom and describe, caused later evolutionary forms of life to assume culture and intelligence, of various sorts.

2.8. Current Disciplines

Before we arrive at the creation or emergence of artificial intelligence, we have to add some newer sciences and technologies that could only exist because our bio-intelligence existed to create them.

Chiefs among these are computer science, cognitive science, and the technologies that both enable and support them. One of the most striking things you’ll notice is that they cross over into one another’s territory. For example, you’ll find linguistics under both computer science and cognitive science. So it's not just disciplines that are converging, but also—and more significantly—the theories and concepts they have studied and shared, sometimes under different names and usually from different perspectives.

In short, the dividing line between biological systems and artificial systems is dissolving before our very eyes.

The following lists are just to give you an idea of the scope of the A.I.-related disciplines.

2.8.1. Computer Science

Computer science is a catch all term for such sub disciplines:

Algorithms, Architecture, Artificial Intelligent, Compression, Computer Engineering, Computational and Applied Mathematics, Computational Mechanics, Computational Learning Theory, Computer Vision, Database, Distributed Computing Aided Design and Manufacturing(CAD/CAM),Formal Methods, Graphics, Handwriting Recognition, Human-Computer Interaction, Information Science, Knowledge Sciences, Linguistics, Logic Programming, Mobile Computing, Modeling Network, Neural Networks, Object-Oriented Programming, Operating Systems, Real-Time Computing, Robotics, Security and Encryption, Software Engineering, Supercomputing and Parallel Computing and Symbolic Computation.

The key areas of study within computer science upon which I propose to focus most attention at this site are those I consider proximate to machine intelligence:

- Connectionist systems (neural networks)
- Rule-based systems (expert systems)
- Case-based reasoning (CBR) systems
- Artificial life
- Genetic programming

2.8.2. Cognitive Science

Like computer science, Cognitive Science defies precise constitutive definition. (It is often referred to in the plural, as "the cognitive sciences".)

"Cognitive Science is one the few fields where modem developments in computer science and artificial intelligence promise to shed light on classical problems in psychology and the philosophy of the mind. Ancient questions of how we see the world, understand language, and reason ,and questions such as 'how a material system can know about the outside world', are being explored with the powerful new conceptual prosthetics of computer modeling".

2.9. Concepts of Artificial Intelligence System

A concept is a number, related either to the memory address where the concept is stored, or to the actual address itself. The contents of this concept is a listing of other numbers (the labels) which are the corresponding (part or concrete) concepts. This number is based on a binary number; a number composed of bits. (A bit is a "binary" data type; that is, it expresses one of only two alternatives. It is a 1 or a 0, a yes or a no, true or false, black or white, something is or is not, yin or yang, voltage or no voltage, an excited nerve or an inhibited nerve. We know that not everything in our world is black or white, but we can still use this binary form of representation by expressing intermediate states, to any desired precision, with a series of bits.)

The reader should note that the label of a concept in an artificial IS does not represent a concept, it is a concept. It is this number (symbol) itself with which the artificial brain

2.10. Summary

In this chapter, one definition says that Artificial Intelligence is the simulation of human intelligence process by machines. The relatively new field of artificial life takes a different approach in attempt to study and understand biological life by synthesizing artificial life forms.

Today it can often be heard that Artificial Intelligence technologies have not lived up to exceptions. It is making its progress in small steps, often invisible to the ordinary observer but very important.

CHAPTER THREE

EXPERT SYSTEMS

3.1. Overview

Certain question and topic come up frequently in the various network discussion groups devoted to and relate to Expert Systems. This file/article is an attempt to gather these questions and their answers into a convenient reference for A.I. researchers, students and practitioners. it is posted on a monthly basis. the hope is that this will cut down on the user time and network and bandwidth used to post, read and respond to the same question over and over, as well as providing education by answering questions some orders ma y not even have thought to ask. Currently this FAQ primarily a list of free and commercial expert system shells, but other questions and answers will be added as they arise.

3.2. Introduction

Knowledge-based expert systems, or simply expert systems, use human knowledge to solve problems that normally would require human intelligence. These expert systems represent the expertise knowledge as data or rules within the computer. These rules and data can be called upon when needed to solve problems. Books and manuals have a tremendous amount of knowledge but a human has to read and interpret the knowledge for it to be used. Conventional computer programs perform tasks using conventional decision-making logic containing little knowledge other than the basic algorithm for solving that specific problem and the necessary boundary conditions. This program knowledge is often embedded as part of the programming code, so that as the knowledge changes, the program has to be changed and then rebuilt. Knowledge-based systems collect the small fragments of human know-how into a knowledge base which is used to reason through a problem, using the knowledge that is appropriate. A different problem, within the domain of the knowledge base, can be solved using the same program without reprogramming. The ability of these systems to explain the reasoning process through back-traces and to handle levels of confidence and uncertainty provides an additional feature that conventional programming doesn't handle.

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. These shells qualify as languages, although certainly with a narrower range of application than most programming languages.

Expert systems technology is one of the most popular and visible facets of **AI.** Expert system shells and knowledge acquisition systems have been developed using disparate approaches to knowledge representation and manipulation and user interfacing.

3.3. Definitions of Expert Systems

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. But in short:

- **Expert systems** (ES's) are programs, usually confined to a specific field that attempt to emulate the behavior of human experts.

3.3.1. 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]

- **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]

- **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].

- **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 away similar to that in which he might consult a human

"Expert, with a similar result." [Edwards and Connell, 1989, p.13]

- **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]

3.4. Advantages and Disadvantages of Expert Systems

This section presents some of the advantages and disadvantages of existing expert systems.

3.4.1. 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.

- If there is a maze of rules (e.g. tax and auditing), then the expert system can "unravel"

the maze.

- **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:

Recency 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.

- **Timeliness** - Fraud and/or errors can be prevented. Information is available sooner for decision making.

- **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.

3.4.2. 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.

3.5. Typical attributes of an Expert system

1. Knowledge is usually represented in declarative form to enable easy reading and modification. Most *ES's* use *IF-THEN* structures for representation; thus, rule-based *ES's* predominate.

2. There is a clear structure to the knowledge representation (excluding neural expert systems).

3. There is a clear distinction between the knowledge representation and the control or manipulation mechanism. Often, the control mechanism is itself rule based (using meta-rules,).

4. A user knowledge-acquisition or knowledge-modification module is often provided for extension of the *ES* (**figure 3.1**).

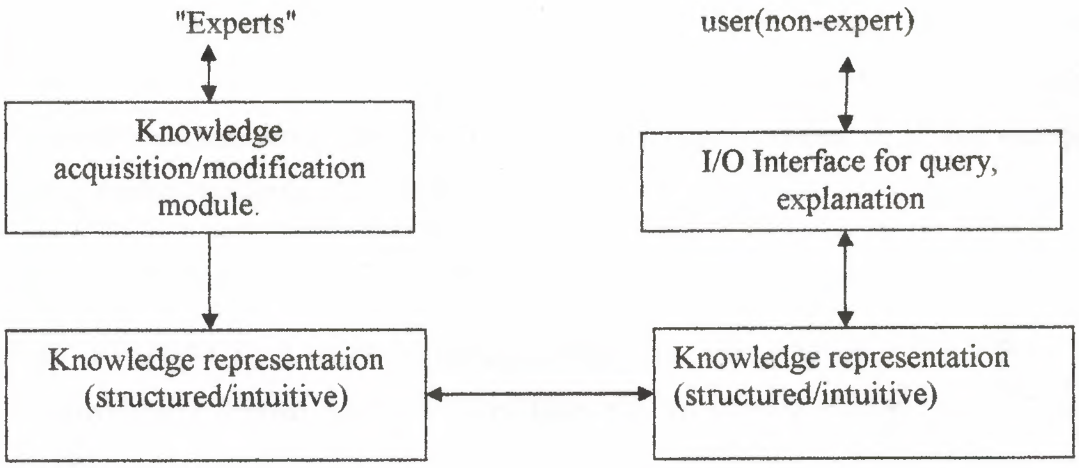

"Experts"
user(non-expert)
Knowledge acquisition/modification module.
I/O Interface for query, explanation
Knowledge representation (structured/intuitive)
Knowledge representation (structured/intuitive)

FIGURE 3.1 structure of typical expert system

3.6. The Appeal of Expert Systems

The development of ES's is motivated by a number of factors, including

- **"Expert"** knowledge is a scarce and expensive resource.
- ES's make expert behavior available to a large audience (i.e., allow more people to

perform like "expert"). This is useful in applications such as system configuration, training, and so forth.

- The integration of the expertise of several experts may lead to ES's that outperform any single expert.

- ES's are not motivated to leave a company for better working conditions, to demand huge salaries (although their development and maintenance costs are often substantial), or to join unions.

3.7. Expert Systems Examples

Approximately 50 recognized ES's are currently in operation. Examples of existing, commercially successful systems are:

1. *XCON* from Digital Equipment Corp., which configures computer systems. *XCON* is written in OPS5.

2. *MACSYMA*, which performs mathematical problem solving using symbolic manipulation rather then numerical evaluation and computation. For example, *MACYMA* is capable of symbolic integration and differentiation of complicated algebraic expressions. Recently, MACSYMA correctly answered all but one question on a freshman calculus final exam at *MIT*.

3. *MYCIN* and *CADUCEUS*, which perform medical diagnosis.

4. *PROSPECTOR*, which guides geological prospecting. When *ROSPECTOR* found a molybdenum deposit worth $100 million, this application gained "respect" [Lernley *1985]*.

5. *CATCH*, which scans photographs to assist police in identifying criminal suspects in New York City.

6. *Dendral,* chemistry – oriented ES for spectroscopic analysis. *Dendral* is one of the oldest ES's, having been develop **(figure 3.2)**

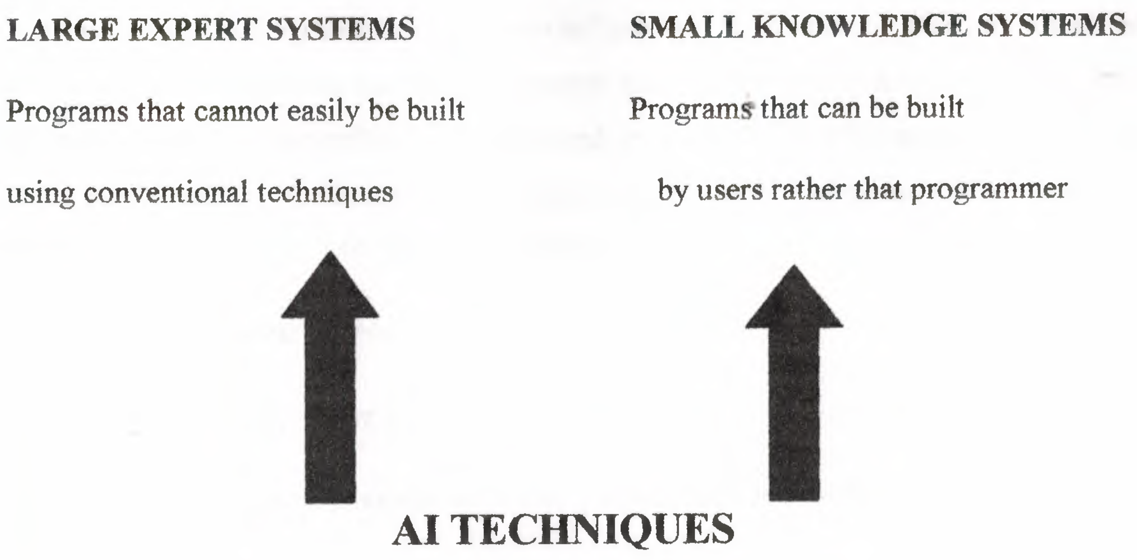

LARGE EXPERT SYSTEMS
Programs that cannot easily be built
using conventional techniques
SMALL KNOWLEDGE SYSTEMS
Programs that can be built
by users rather that programmer
AI TECHNIQUES

FIGURE 3.2

The major trends in AI applications.

3.8. Expert System Limitations

One might expect the performance of expert systems, which could tirelessly and exhaustively consider every possibility associated with a problem, to outperform humans in a spectrum of applications. This is currently not the case. ES developers have discovered that knowledge acquisition can be show and expensive. Furthermore, systems tend to be "brittle" in the sense that slight modifications in the application lead to unacceptable deviations in ES performance. It is not incidental that a human spends approximately 12 years past the age of 5 (or thereabouts) in formal schooling. Notwithstanding the possible lack of efficiency in this process, a significant amount of both information and experience (which is perhaps not as easily quantifiable) is gained over this time interval. In addition, most perceived experts have a considerable amount of additional informal and formal education past his point.

Thus, we should not be surprised at even the practical difficulty of representing expert behavior.

From the definition of expert systems did not include the term reason. Unfortunately, human experts do not reach conclusions solely through the application of a describable reasoning process. In fact, many experts attribute their success in reaching (what are proven over time to be) correct conclusion to "gut fee!" or "instinct." This implies either a significant experience or that defies apparent quantification. This helps to explain why ES application has heretofore been restricted to a narrow lack of success of ES's in other areas, notably stock market forecasting, detective (criminal investigation) work, and "football picks," provides a challenge for the future.

3.9. Expert System Development

The first questions an expert system developer must ask are the following:

1. Are bona fide experts available whose performance is significantly better that that of amateurs?

2. Can their expertise be automated?

3. Does it make practical and economic sense to develop an ES?

3.9.1. Experts Query (The Role of "Knowledge Engineers")

The development of expert systems involves consultation of an expert (or group of experts) with the aim of developing a manipulability knowledge base. Thus, the first phase of the process consists of the information of a database of domain-specific knowledge. In the expert integration process, the formulation of "good" questions is paramount. Fortunately, experts often phrase problem-solving methodologies in term of IF-THEN structures.

For example, if it looks like a duck, talks like a duck... then I classify it a duck...

ES VERIFIVATION. The development of an expert system is almost always an iterative task, involving the cycle of expert query, database formation, development of the inference strategy, verification of system performance, and so on. This design process is shown in **(Fig. 3.3)** The gap between the concepts of an ES and a finished, delivered product may be enormous. The necessary application-specific selection of a reasoning structure, interviewing of experts, and development of a prototype, refinement, user training, and documentation may take several years.

One of the most important aspects of ES development is verification of system operation. A set of test cases is developed and used by both the ES and the human expert; when responses differ, modifications to the system are identified and implemented. To be useful, the system should provide good user interaction as well. A response such as Patient has disease *x*, is probably insufficient; even if it is correct, since no explanation of the inference process is provided. An explanation may be as simple as indicating the sequence of rules used, or may be as complicated as indicating all possible inference paths considered and the logic that indicates the most appropriate. Note that

1. Some measure of confidence or preference in the logic of (competing) paths may be desired. This is the subject of a later section.

2. Rules are often augmented to include a BECAUSE field or descriptor, which serves to further explain the rule. The system explanation could then consist of the output of the BECAUSE statement, in the order in which rules were used. (Shown in **figure 3.3**).

3.10. How Different is an Expert Systems?

1. Implementation Level

Expert systems are systems, which are implemented with production rule. On the one hand, this was the answer with the highest rating when asked users of expert system technology. On the other hand, this answer does not reflect the state of the art currently, there exist several alternative implementation formalism and sometimes expert systems are written in standard programming languages like C++ or Smalltalk to improve efficiency, portability, or integration in environments. Therefore, these systems would no longer an expert system.

2. Design Level

Expert systems are systems with a specific architecture. I think that this answer reflects a higher level of expert system development because it abstracts from implementation details, i.e. the used implementation formalism. One answer II (often) found in the literature is that an expert system is a system, which consists of the following components: a knowledge base, an interpreter, a user interface, a knowledge acquisition component, and an explanation component.

3. Specification Level

Expert systems are systems, which solve specific types of tasks. As a reaction to the so-called software crisis in the late sixties the development of process models with a separation of specification, design, and implementation of a system arose in the software engineering community. Examples are the waterfall model or the spiral model. Currently, a similar development can be observed for expert systems. Therefore, distinctions, which concern design or implementation, are no longer sufficient to determine the specific character of expert systems. Otherwise, during the specification process (the knowledge acquisition or modeling process) one could not fix a difference to standard software. Therefore it is necessary to look for differences, which refer to the functionality of a system. I could find two complementary answers in the literature: First, expert systems (or knowledge-based systems) solve tasks with less algorithmic complexity but which require a high amount of domain and task specific knowledge.

4. The answer of a sociologist

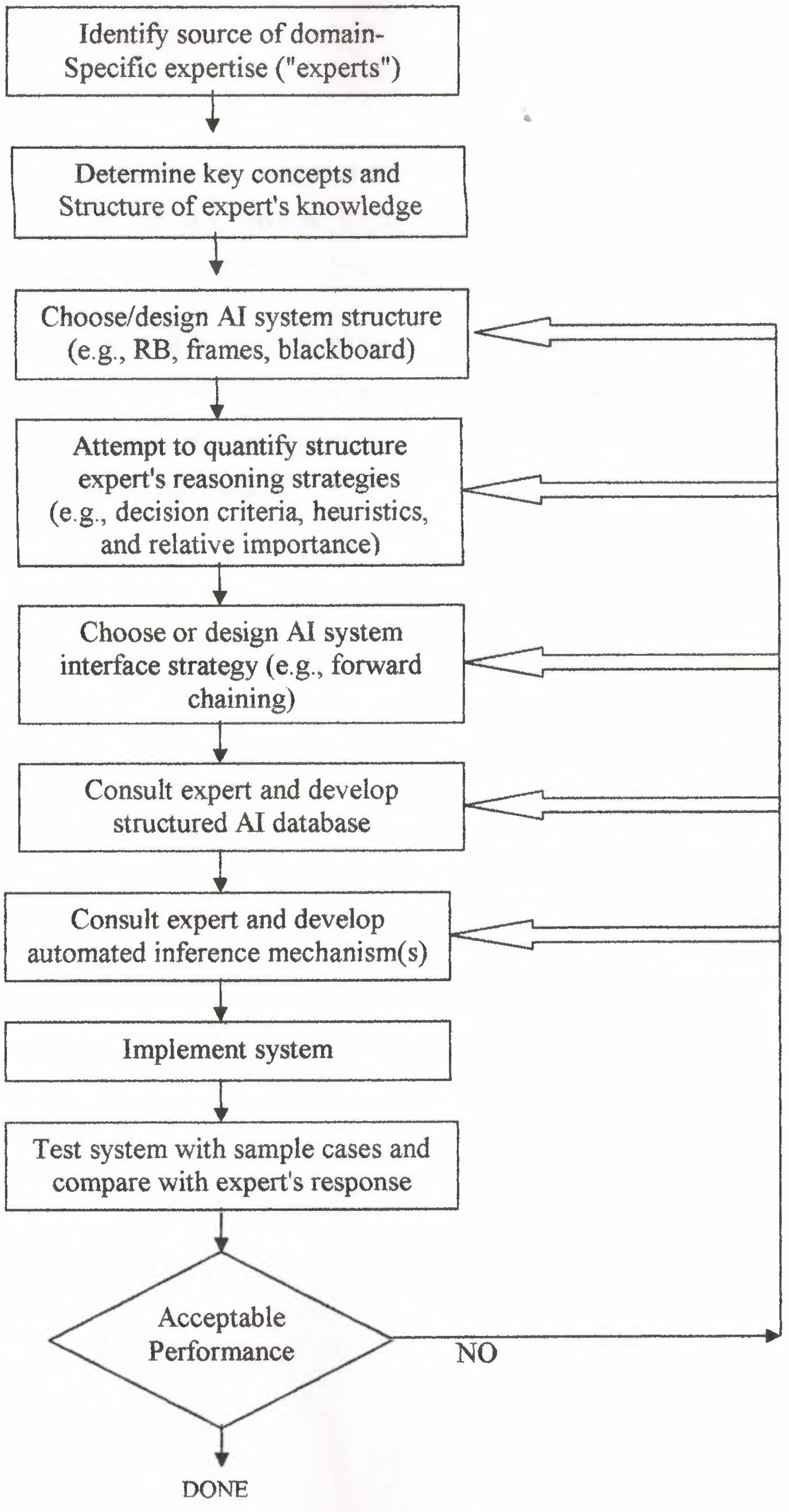
A fourth answer (maybe the one closest to truth) says that expert system is just the name for application systems, which are built by members of the AL-community. But is this the only significant difference? I am looking forward for your opinions and arguments!

3.11. Making the Expert System Easy to Use

Whether or not an expert system achieves success may be determined by the nature of its user interface. This is the part of the expert system that interacts with the user. Even the most powerful expert system will not be applied if it requires too much effort on the part of the user. For this reason, it is important to make the computer as easy for the user to operate as possible. Almost all modern development programs offer the capacity to interact with the system through both text and graphics.

3.12. Developing an Expert System

The power of an expert system is derived from the knowledge of the expert. The lack of computer skills should not inhibit anyone from using this tool in implementing their disease management program. There are published procedures that one can follow to develop expert systems, although rigid adherence to these procedures is not necessarily a prerequisite for a successful development effort. Match your development technique with your own style of thinking and your resources and problem area. Listed below are several examples of where the developers of the Penn State Apple Orchard Consultant



Identify source of domain-Specific expertise ("experts")
Determine key concepts and Structure of expert's knowledge
Choose/design AI system structure (e.g., RB, frames, blackboard)
Attempt to quantify structure expert's reasoning strategies (e.g., decision criteria, heuristics, and relative importance)
Choose or design AI system interface strategy (e.g., forward chaining)
Consult expert and develop structured AI database
Consult expert and develop automated inference mechanism(s)
Implement system
Test system with sample cases and compare with expert's response
Acceptable Performance
NO
DONE

FIGURE 3.3(system explanation)

3.13. Summary

There are expert systems that can diagnose human illnesses, make financial forecasts, and schedule routes for delivery vehicles. Some expert systems are designed to take the place of human experts, while others are designed to aid them.

Expert systems are part of a 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.

CHAPTER FOUR

NEURAL NETWORKS

4.1. Overview

Rather than using a digital model, in which all computations manipulate zeros and ones, a neural network works by creating connections between processing elements, the computer equivalent of neurons. The organization and weights of the connections determine the output.

Neural networks are particularly effective for predicting events when the networks have a large database of prior examples to draw on. Strictly speaking, a neural network implies a non-digital computer, but neural networks can be simulated on digital computers.

The field of neural networks was pioneered by Bernard Widrow of Stanford University in the 1950s. To date, there are very few commercial applications of neural networks, but the approach is beginning to prove useful in certain areas that involve recognizing complex patterns, such as voice recognition.

4.2. Introduction to Neural Networks

A very simple neural network was developed to predict the number of runs scored by a baseball team in a game based on total team offensive statistics. The resulting model could then be used to:

Compare the contribution of players to team run production based on individual statistics. Determine the key statistics and their relative importance in run production. Better identify the worth of individual players to the team for the purpose of supporting salary arbitration arguments. The resulting neural network results were compared to a linear regression model's results. A regression model is a standard statistical tool that was use as a basis of comparison in model performance. The same data were used to develop and test both models. Game totals of the key offensive statistics such as hits, home runs and base on balls were used. The data used for the model development consisted of a small set of *135* baseball games obtained from a realistic computer simulation based upon actual 1992 major league statistics. Both teams' data were used giving a dataset of 264 observations. A true measure of the complexity of the problem would require thousands of observations. This was beyond the resources available this investigation. However, the number of observations available does make for an instructive problem.

Neural Networks are the result of academic investigations that involve using mathematical formulations to model nervous system operations. The resulting techniques are being successfully applied in a variety of everyday business applications. Neural Networks represent a meaningfully different approach to using computers in the workplace. A neural network is used to learn patterns and relationships in data. The data may be the results of a market research effort; the results of a production process given vary operational conditions, or the decisions of a loan officer given a set of loan applications. Regardless of the specifics involved, applying a neural network is a substantial departure from traditional approaches. Traditionally a programmer or an analyst specifically "codes" every facet of the problem in order for the computer to "understand" the situation. Neural networks do not require the explicit coding of the problem. For example, to generate a model that performs sales forecast, a neural network only needs to be given raw data related to the problem. The raw data might consist of: history of past sales, prices, competitors' prices, and other economic variables. The neural network sorts through this information and produces an understanding of the factors impacting sales. The model can then be called upon to provide a prediction of future sales given a forecast of the key factors.

These advancements are due to the creation of neural network learning rules, which are the algorithms used to "learn" the relationships in the data. The learning rules enable the network to "gain knowledge" from available data and apply that knowledge to assist a manager in marking key decisions.

4.3 What is a Neural Network?

A machine is designed to model the way in which brain performs a particular task on function of interest. The NN is usually implemented using electronic components on simulated in software on a digital computer.

In summary is the massively parallel-distributed process that has a natural propensity for strong experiential knowledge and making it available for use.

Neural Networks are a different paradigm for computing:

- Von Neumann machines are based on the processing/memory abstraction of human information processing.
- Neural networks are based on the parallel architecture of animal brains.

Neural networks are a form of multiprocessor computer system, with

- Simple processing elements.
- A high degree of interconnection.
- Simple scalar messages.
- Adaptive interaction between elements.

A biological neuron may have as many as 10,000 different inputs, and may send its output (the presence or absence of a short-duration spike) to many other neurons. Neurons are wired up in a 3-dimensional pattern.

Real brains, however, are orders of magnitude more complex than any artificial neural network so far considered.

Example: A simple single unit adaptive network:

The network has 2 inputs, and one output. All are binary. The output is

1 if $\mathbf{W}_0 * \mathbf{I}_0 + \mathbf{W}_1 * \mathbf{I}_1 + \mathbf{W}_b > 0$

0 if $\mathbf{W}_0 * \mathbf{I}_0 + \mathbf{W}_1 * \mathbf{I}_1 + \mathbf{W}_b <= 0$

We want it to learn simple OR: output 1 if either I_0 or I_1 is 1.

4.4. Artificial Neurons and How They Work

The fundamental processing element of a neural network is a neuron. This building block of human awareness encompasses a few general capabilities. Basically, a biological neuron receives inputs from other sources, combines them in some way, performs a generally nonlinear operation on the result, and then outputs the final result Fig. 4.1 shows the relationship of these four parts.

Within humans there are many variations on this basic type of neuron, further complicating mans attempts at electrically replicating the process of thinking. Yet, all natural neurons have the same four basic components. These components are known by their biological names - dendrites, soma, axon, and synapses. Dendrites are hair-like extensions of the soma, which act like input channels. These input channels receive their input through the synapses of other neurons. The soma then processes these Incoming signals over time. The soma then turns that processed value into an output, which is sent out to other neurons through the axon and the synapses.

Recent experimental data has provided further evidence that biological neurons are structurally more complexes than the simplistic explanation above. They are significantly more complex than the existing artificial neurons that are built into today's artificial neural networks. As biology provides a better understanding of neurons, and as technology advances, network designers can continue to improve their systems by building upon man's understanding of the biological brain.

But currently, the goal of artificial neural networks is not the grandiose recreation of the brain. On the contrary, neural network researchers are seeking an understanding of nature's capabilities for which people can engineer solutions to problems that have not been solved by traditional computing.

To do this, the basic unit of neural networks, the artificial neurons simulates the four basic functions of natural neurons. Fig. 4.2 shows a fundamental representation of an artificial neuron.

In Fig. 4.2, various inputs to the network are represented by the mathematical symbol, x (n). Each of these inputs is multiplied by a connection weight. These weights are represented by w (n). In the simplest case, these products are simply summed, fed through a transfer function to generate a result, and then output. This process lends itself to physical implementation on a large scale in a small package. This electronic implementation is still possible with other network structures, which utilize different summing functions as well as different transfer functions.

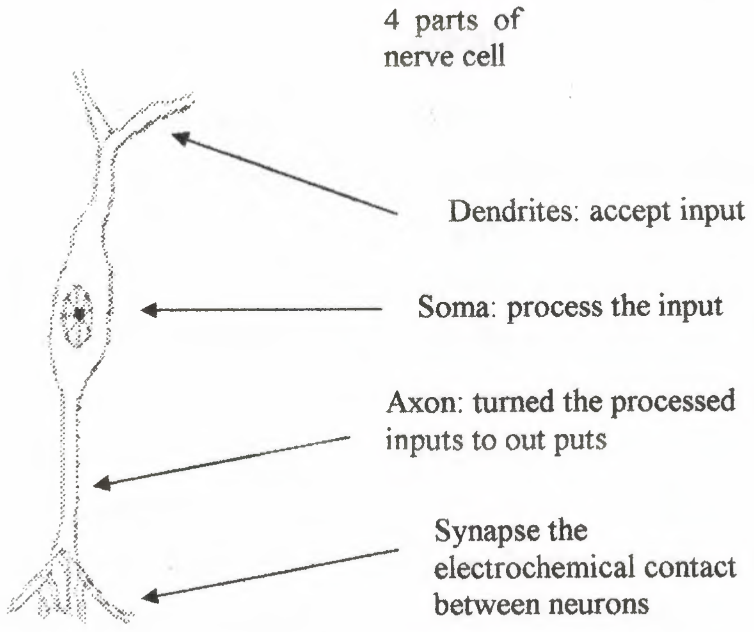

4 parts of nerve cell
Dendrites: accept input
Soma: process the input
Axon: turned the processed inputs to out puts
Synapse the electrochemical contact between neurons

Figure 4.1 Simple Neuron

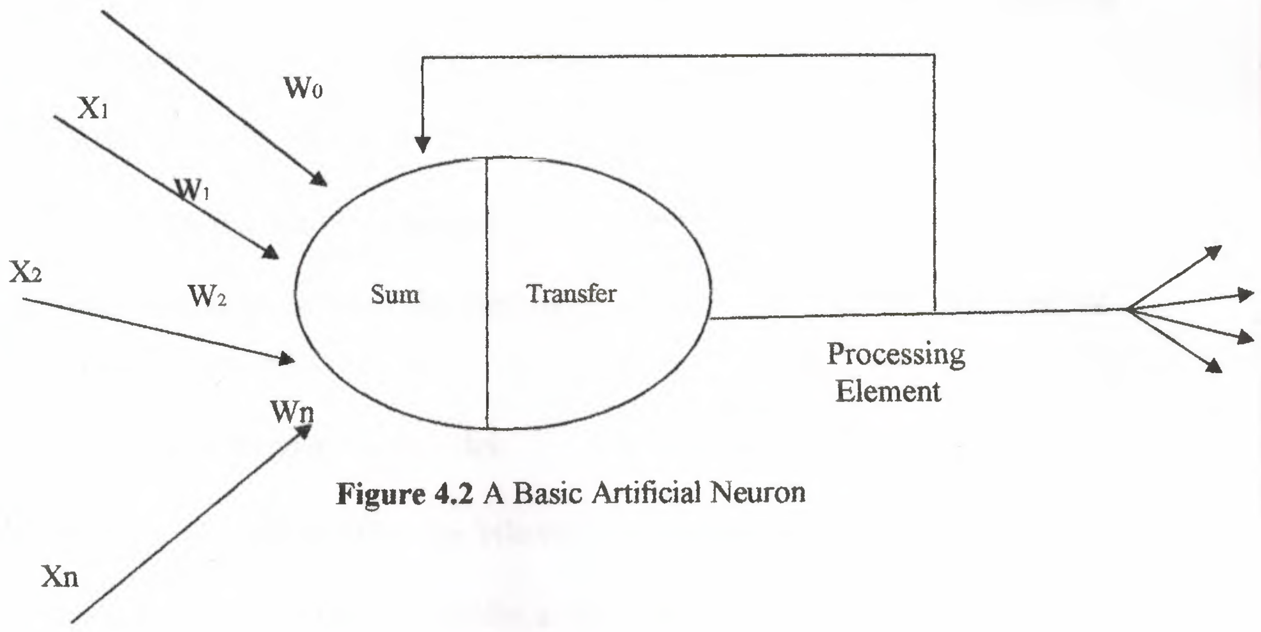


Figure 4.2 A Basic Artificial Neuron

Some applications require "black and white," or binary, answers. These applications include the recognition of text, the identification of speech, and the image deciphering of scenes. These applications are required to turn real-world inputs into discrete values. These potential values are limited to some known set, like the ASCII characters or the most common 50,000 English words. Because of this limitation of output options, these applications don't always utilize networks composed of neurons that simply sum up, and thereby smooth, inputs. These networks may utilize the binary properties of ORing and ANDing of inputs. These functions, and many others, can be built into the summation and transfer functions of a network.

Other networks work on problems where the resolutions are not just one of several known values. These networks need to be capable of an infinite number of responses. Applications of this type include the "intelligence" behind robotic movements. This "intelligence" process inputs and then creates outputs, which actually cause some device to move. That movement can span an infinite number of very precise motions. These networks do indeed want to smooth their inputs which, due to limitations of sensors, come in non-continuous bursts, say thirty times a second. To do that, they might accept these inputs, sum that data, and then produce an output by, for example, applying a hyperbolic tangent as a transfer functions. In this manner, output values from the network are continuous and satisfy more real world interfaces.

Other applications might simply sum and compare to a threshold, thereby producing one of two possible outputs, a zero or a one. Other functions scale the outputs to match the application, such as the values minus one and one. Some functions even integrate the input data over time, creating time-depend networks.

The NN resembles the brain in two respects:

- Knowledge is acquired by network through a learning process.
- Interconnections between neurons have strength known as synaptic weights. These are used to store knowledge.

4.5. Benefits of Neural Networks

The use of neural networks offers the following useful properties and capabilities:

1. Nonlinearity: A neuron is basically a nonlinear device. Consequently, a neural network, made up of an interconnection of neurons is itself nonlinear.

2. Input –output mapping: Popular paradigm of learning called supervised learning involves the modification of the synaptic weights of a neural network by applying a set of labeled training samples or task examples.

3. Adaptively: Neural networks have a built-in capability to adapt their synaptic weights to changes in the surrounding environment.

4. Evidential response: In the context of pattern classification, a neural network can be designed to provide information may be used to reject ambiguous patterns, should they arise, and thereby improve the classification performance of the network.

5. Contextual information: Knowledge is represented by the very structure and activation state of a neural network.

4.5. Supervised and Unsupervised Training

Neural Networks can be classified according to the way of learn. Learning can be performed on as a supervised or unsupervised basis.

Once a network has been structured for a particular application, that network is ready to be trained. To start this process the initial weights are chosen randomly. Then, the training, or learning begins.

There are two approaches to training - supervised and unsupervised. Supervised training involves a mechanism of providing the network with the desired output either by manually 'grading' the network's performance or by providing the desired outputs with the inputs. Unsupervised training is where the network has to make sense of the inputs without outside help.

The vast bulk of networks utilize supervised training. Unsupervised training is used to perform some initial characterization on inputs. However, in the full-blown sense of being truly self-learning, it is still just a shining promise that is not fully understood, does not completely work, and thus is relegated to the lab.

4.5.1. Supervised Training

In supervised training, both the inputs and the outputs are provided. The network then processes the inputs and compares its resulting outputs against the desired outputs. Errors are then propagated back through the system, causing the system to adjust the weights, which control the network. This process occurs over and over as the weights are continually tweaked. The set of data, which enables the training, is called the "training set." During the training of a network the same set of data is processed many times as the connection weights are ever refined.

The current commercial network development packages provide tools to monitor how well an artificial neural network is converging on the ability to predict the right answer. These tools allow the training process to go on for days, stopping only when the system reaches some statistically desired point, or accuracy. However, some networks never learn. This could be because the input data does not contain the specific information from which the desired output is derived. Networks also don't converge if there is not enough data to enable complete learning. Ideally, there should be enough data so that part of the data can be held back as a test. Many layered networks with multiple nodes are capable of memorizing data. To monitor the network to determine if the system is simply memorizing its data in some no significant way, supervised training needs to hold back a set of data to be used to test the system after it has undergone its training. (Note: memorization is avoided by not having too many processing elements.) If a network simply can't solve the problem, the designer then has to review the input and outputs, the number of layers, the number of elements per layer, the connections between the layers, the summation, transfer, and training functions, and even the initial weights themselves. Those changes required to create a successful network constitute a process wherein the "art" of neural networking occurs.

Another part of the designer's creativity governs the rules of training. There are many laws (algorithms) used to implement the adaptive feedback required to adjust the weights during training. The most common technique is backward-error propagation, more commonly known as back-propagation. These various learning techniques are explored in greater depth later in this report.

Yet, training is not just a technique. It involves a "feel," and conscious analysis, to insure that the network is not overstrained. Initially, an artificial neural network configures itself with the general statistical trends of the data. Later, it continues to "learn" about other aspects of the data, which may be spurious from a general viewpoint.

When finally the system has been correctly trained, and no further learning is needed, the weights can, if desired, be "frozen." In some systems this finalized network is then turned into hardware so that it can be fast. Other systems don't lock themselves in but continue to learn while in production use.

4.5.2. Unsupervised or Adaptive Training

The other type of training is called unsupervised training. In unsupervised training, the network is provided with inputs but not with desired outputs. The system itself must then decide what features it will use to group the input data. This is often referred to as self-organization or adaptation.

At the present time, unsupervised learning is not well understood. This adaptation to the environment is the promise, which would enable science fiction types of robots to continually learn on their own as they encounter new situations and new environments. Life is filled with situations where exact training sets do not exist. Some of these situations involve military action where new combat techniques and new weapons might be encountered. Because of this unexpected aspect to life and the human desire to be prepared, there continues to be research into, and hope for, this field. Yet, at the present time, the vast bulk of neural network work is in systems with supervised learning. Supervised learning is achieving results.

One of the leading researchers into unsupervised learning is Tuevo Kohonen. He has developed a self-organizing network, sometimes called an auto-associator, that learns without the benefit of knowing the right answer. It is an unusual looking network in that it contains one single layer with many connections. The weights for those connections have to be initialized and the inputs have to be normalized. The neurons are set up to compete in a winner-take-all fashion.

Kohonen continues his research into networks that are structured differently than standard; feed forward, back-propagation approaches. Kohonen's work deals with the grouping of neurons into fields. Neurons within a field are "topologically ordered." Topology is a branch of mathematics that studies how to map from one space to another without changing the geometric configuration. The three-dimensional groupings often found in mammalian brains are an example of topological ordering.

Kohonen has pointed out that the lack of topology in neural network models make today's neural networks just simple abstractions of the real neural networks within the brain. As this research continues, more powerful self-learning networks may become possible. But currently, this field remains one that is still in the laboratory.

4.6. Applications of Neural Networks

Neural networks, inspired by the information-processing strategies of the brain, are proving to be useful in a variety of applications, and their full potential is far from realization. In manufacturing, for example, neural network technology is being increasing applied or planned for application in complex manufacturing processes that have not been adequately tackled by more conventional technologies.

The features equip the neural network very well for modeling, analyzing, forecasting, and optimizing the performance of manufacturing plants. The possibility for exploiting the correlating and generalizing strengths of neural networks in the computer-integrated manufacturing of integrated circuits has been highlighted in a recent survey that underscores the increasing interest being shown by industries and universities to apply neural networks to manufacturing tasks of very great complexities, including process modeling, process optimization, monitoring and control, process diagnosis, and commercial product fabrication. The processes themselves could have diver's origins in civil, chemical, electrical, industrial, materials, mechanical, nuclear, and transportation engineering.

Neural networks have been extensively in continuous speech recognition and synthesis, image processing and coding, pattern recognition and classification, power load forecasting, interpretation and prediction of financial trends for stock market analysis, manufacturing of composite structures, production of printed circuit boards, process modeling, monitoring and control, reliable and flexible routing of telecommunication networks in the presence of component failure or malfunction, and many other problems occurring in our expanding technological base. The advantages of the neural network in many of these applications are that the network can be updated continuously with new data to optimize its performance at any instant; the network's ability to handle a large number of input variables rapidly; and the network's ability to filter noisy data and interpolate incomplete data.

Though the domain of application of artificial neural networks is extensive and expanding, the degree of success has varied with the type of the application. This does not close the door to possibilities for improvement but instead challenges us to examine the limitations in our approaches with respect to the choice of structures and rules.

Because all artificial neural networks are based on the concept of neurons, connections and transfer functions, there is a similarity between the different structures or architectures or neural networks. The majority of the variations stems from the various learning rules and how those rules modify a network's typical topology. The following sections outline some of the most common artificial neural networks. They are organized in very rough categories of application. These categories are not meant to be exclusive, they are merely meant to separate out some of the confusion over network architectures and their best matches to specific applications.

Basically, most applications of neural networks fall into the following five categories:

1. prediction

2. classification

3. data association

4. data conceptualization

4.6.1. New Application Areas

Pen pc's: where one can write on a table and the writing will be recognized and translated into ASCII text.

Speech and vision recognition systems: not new but neural networks are becoming increasingly part of such systems. They are used as a system component, in conjunction with traditional computers.

White goods and toys: as neural network chips become available, the possibility of simple cheap systems, which have learned to recognize simple entities (e.g. walls looming, or simple commands like Go, or Stop), may lead to their incorporation in toys and washing machines etc. already the Japanese are using a related technology, fuzzy logic, in this way. There is considerable interest in the combination of fuzzy and neural technologies.

4.7. Where are Neural Networks Going?

A great deal of research is going on in neural networks worldwide. This ranges from basic research into new and more efficient learning algorithms, to networks, which can respond to temporally varying patterns, to techniques for implementing neural networks directly in silicon. Already one chip commercially available exists, but it does not include adaptation and is working on the learning problem. Production of a learning chip would allow the application of this technology to a whole range of problems where the price of a PC and software cannot be justified. There is particular interest in sensory and sensing applications: nets, which learn to interpret real-world sensors and learn about their environment.

4.8. How Neural Networks Differ from Traditional Computing and Expert Systems

Neural networks offer a different way to analyze data, and to recognize patterns within that data, than traditional computing methods. However, they are not a solution for all computing problems. Traditional computing methods work well for problems that can be well characterized. Balancing checkbooks, keeping ledgers, and keeping tabs of inventory are well defined and do not require the special characteristics of neural networks. Table 4.1 identifies the basic differences between the two computing approaches.

Traditional computers are ideal for many applications. They can process data, track inventories, network results, and protect equipment. These applications do not need the special characteristics of neural networks.

Expert systems are an extension of traditional computing and are sometimes called the fifth generation of computing. (First generation computing used switches and wires. The second generation occurred because of the development of the transistor. The third generation involved solid-state technology, the use of integrated circuits, and higher level languages like COBOL, FORTRAN, and "C". End user tools, "code generators," are known as the fourth generation.) The fifth generation involves artificial intelligence.

Typically, an expert system consists of two parts, an inference engine and a knowledge base. The inference engine is generic. It handles the user interface, external files, program access, and scheduling.

TABLE 4.1 Comparison of Computing Approaches

Characteristics	**Traditional Computing (including expert system)**	**Artificial Neural Network**
Processing style function	Sequential logically via rules Concepts calculations	Parallel via images Pictures controls
Learning method application	By rules accounting, word processing, math inventory, digital communications	By example sensor processing, speech recognition, pattern recognition text recognition

The knowledge base contains the information that is specific to a particular problem. This knowledge base allows an expert to define the rules, which govern a process. This expert does not have to understand traditional programming. That person simply has to understand both what he wants a computer to do and how the mechanism of the expert system shell works. It is this shell, part of the inference engine, which actually tells the computer how to implement the expert's desires. This implementation occurs by the expert system generating the computer's programming itself; it does that through "programming" of its own. This programming is needed to establish the rules for a particular application. This method of establishing rules is also complex and does require a detail-oriented person.

Efforts to make expert systems general have run into a number of problems. As the complexity of the system increases, the system simply demands too much computing resources and becomes too slow. Expert systems have been found to be feasible only when narrowly confined.

Artificial neural networks offer a completely different approach to problem solving and they are sometimes called the sixth generation of computing. They try to provide a tool that both programs itself and learns on its own. Neural networks are structured to provide the capability to solve problems without the benefits of an expert and without the need of programming. They can seek patterns in data that no one knows are there. A comparison of artificial intelligences expert systems and neural networks is contained in Table 4.2.

Expert systems have enjoyed significant successes. However, artificial intelligence has encountered problems in areas such as vision, continuous speech recognition and synthesis, and machine learning. Artificial intelligence also is hostage to the speed of the processor that it runs on. Its is restricted to the theoretical limit of a single processors. Artificial intelligence is also burdened by the fact that experts don't always speak in rules.

Yet, despite the advantages of neural networks over both expert systems and more traditional computing in these specific areas, neural nets are not complete solutions. They offer a capability that is not ironclad, such as a debugged accounting system.

They learn, and as such, they do continue to make "mistakes."

Furthermore, even when a network has been developed, there is no way to ensure that the network is the optimal network.

Neural systems do exact their own demands. They do require their implementer to meet a number of conditions. These conditions include:

- A data set which includes the information, which can characterize the problem.
- An adequately sized data set to both train and test the network.
- An understanding of the basic nature of the problem to be solved so that basic first-cut decision on creating the network can be made. These decisions include the activation and transfer functions, and the learning methods.
- An understanding of the development tools.
- Adequate processing power (some applications demand real-time processing that exceeds what is available in the standard, sequential processing hardware. The development of hardware is the key to the future of neural networks).

Once these conditions are met, neural networks offer the opportunity of solving problems in an arena where traditional processors lack both the processing power and a step-by-step methodology. A number of very complicated problems cannot be solved in traditional computing environments. For example, speech is something that all people can easily parse and understand.

Table 4.2 Comparisons of Expert Systems and Neural Network

Characteristics	**Architecture used for expert system**	**Artificial Neuron**
Processors	VLSI(traditional processor)	Artificial Neural Network; variety of technologies; hardware development.
Processing approach	Separate, process problem rule at a one time; sequential.	The same, multiple
Connections	Connections	Dynamically self programming
Self learning	Only algorithmic parameters modified	Continuously adaptable
Fault tolerance	Non without special processor	Significant in the very nature of the interconnected
Programming	Through a rule based complicated	Self programming ; but network must be setup properly
Ability to be fast	Require a big processors	Require multiple custom-built chips
Significant in the very nature of the interconnected	None	Moderate

A person can understand a southern drawl, a Bronx accent, and the slurred words of a baby. Without the massively paralleled processing power of a neural network, this process is virtually impossible for a computer. Image recognition is another task that a human can easily do but which stymies even the biggest of computers. A person can recognize a plane as it turns, flies overhead, and disappears into a dot. A traditional computer might try to compare the changing images to a number of very different stored patterns.

4.9. Summary

The new way of computing requires skills beyond traditional computing. It is a natural evolution. Initially, computing was only hardware and engineers made it work. Then, there were software specialists - programmers, systems engineers, data base specialists, and designers. Now, there are also neural architects. This new professional needs to be skilled different than his predecessors of the past. For instance, he will need to know statistics in order to choose and evaluate training and testing situations. This skill of making neural networks work is one that will stress the logical thinking of current software engineers.

In summary, neural networks offer a unique way to solve some problems while making their own demands. The biggest demand is that the process is not simply logic. It involves an empirical skill, an intuitive feel as to how a network might be created.

CHAPTER FIVE

GENETIC ALGORITHMS

5.1. Overview

After scientists became disillusioned with classical and neo-classical attempts at modeling intelligence, they looked in other directions. Two prominent fields arose, connectionism (neural networking, parallel processing) and evolutionary computing. Genetic algorithms are a part of evolutionary computing, which is a rapidly growing area of artificial intelligence. As you can guess, genetic algorithms are inspired by Darwin's theory about evolution. Simply said, solution to a problem solved bye genetic algorithms is evolved.

5.2. History of Genetic Algorithms

Algorithm is started with a set of solutions (represented by chromosomes) called population. Solutions from one population are taken and used to form a new population. This is motivated by a hope, that the new population will be better than the old one. Solutions which are selected to form new solutions (offspring) are selected according to their fitness-the more suitable they are the more chances they have to reproduce.

At Genetic Algorithms we study nature's search algorithm of choice, genetics and evolution, as a practical approach to solving difficult problems on a computer. Genetic algorithms (G.As) and evolutionary computation have been around since the cybernetics movement of 1950s, but they have undergone a renaissance since the mid-1980sto the point where many walks of human endeavor are benefiting from this approach. For example, problems as different as jet engine design, electromagnetic antenna-absorber optimization and design, analog and logic electronic circuit synthesis, structural optimization and layout, factory and project scheduling, control system synthesis, music composition, image recognition, and automated programming have been successfully tackled.

The mechanics of a genetic algorithm (G.A.) are conceptually simple:

Maintain a population of solutions coded as artificial chromosomes.

Select the better solutions for recombination (crossover) of the mating chromosomes.

Perform mutation and other variation operators on the chromosomes.

Use these offspring to replace poorer solutions or to create a new generation altogether.

Theory and empirical results demonstrate that G.As can be guaranteed to solve a broad class of provably hard problems, quickly, reliably, and accurately.

Idea of evolutionary computing was introduced in the 1960s by Rothenberg in his work "Evolution strategies" (Evolutions strategy in original). His idea was then developed by other researchers. Genetic Algorithms (GAs) were invented by John Holland and developed by him and his students and colleagues. This lead to Holland's book "Adoptions in Natural and Artificial Systems" published in 1975.

In 1992 John Koza has used genetic algorithm to evolve programs to perform certain tasks. He called his method "genetic programming" (GP). LISP programs were used, because programs in this language can express in the form of a "parse tree", which is the object the GA works on.

Unlike natural evolution, genetic algorithms do not require millions of years to produce results. However, the system may need to run for many hours or even days. Large, complex problems require a fast computer in order to obtain good solutions in a reasonable amount of time. Mining of large datasets by genetic algorithms has only recently become practical due to the availability of affordable high-speed computers such as the DEC Alpha.

In the 1950s and 1960s several computer scientists independently studied evolutionary systems with the idea that evolution could be used as optimization tool for engineering problems. The idea in all these systems was to evolve a population of candiate solution to a given problem, using operator inspired by natural genetic variation and natural selection.

In the 1960s Rechenberg [10] (1965,1973) introduce "evolutionary strategies", a method he used to optimize real valued parameters for devices such as airfoils. This idea was further developed by Schwefel [11] (1675,1977). The field of evolution of strategies has remained an active area of research, mostly developing independently from the field of genetic algorithms.

Several other people working in 1950s and the 1960s developed evolution inspired algorithms for optimization and machine learning. Box (1957), Friedman (1959), Bledsoe (1962), Bermermann (1962), and Reed, Toombs, and Baricelli [12] (1967) all worked in this area, though their work has been given little or none attention or follow up that evolution strategies, evolutionary programming, and genetic algorithms have seen.

Genetic algorithms were invented by John Holland [13] in 1960s and were developed by Holland and his students and colleagues at the University of Michigan in 1960s and 1970s. In contrast with evolution strategies and evolutionary programming, Holland's original goal was not to design algorithms to solve specific problems, but rather to formally study the phenomena of adaptation as it occurs in nature and to develop ways in which the mechanism of natural adaptation might be imported into computer system.
In the last several years there has been widespread interaction researchers studying various evolutionary computation methods, and the boundaries between Genetic Algorithms, evolutionary strategies, evolutionary programming, and other evolutionary approaches have been broken down to some extent.

5.3. What is Genetic Algorithms?

The G.A. is a model of machine learning, which derives its behavior from a metaphor of the processes of *evolution* in nature. This is done by the creation within a machine of a *population* of Individuals represented by Chromosomes, in essence a set of character string that is analogous to the base-4 chromosomes that we see in our own DNA. The individuals in the population then go through a process of evolution.

We should note that *evolution* (in nature or anywhere else) is not a purposive or directed process. That is, there is no evidence to support the assertion that the goal of evolution is to produce mankind. Indeed, the processes of nature seem to boil down to different individuals competing for resources in the environment. Some are better than others. Those are more likely to survive and propagate their genetic material.

5.4. How Do Genetic Algorithms Work?

Here there are some explanations about the G.A. based on the relevant sections in "An Introduction to Genetic Algorithms" by Melanie Mitchell. It contains a great deal of mathematics, algebra, and logic is advised.

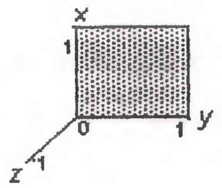
5.4.1. Schemata and Other Terminology

The pioneer of genetic algorithms (GA), John Holland, was the first person to fully describe GAs mathematically. GAs are designed to search for, discover, emphasize and breed good solutions (or "building blocks") to a problem in a highly parallel fashion. Holland invented the idea of *schemas* (or *schemata*) to formally conceptualize the notion of "building blocks."

Each schema is a template for a bit string of length *l*. The schemata are comprised of one's and zero's (defining the bits), and asterisk ('*') that act as wild cards within the string. Therefore, the schema:

H = 1 * * * 0 *

represents all bit strings that begin with one and have a '0' at the 5th bit position. Examples of bit strings that satisfy this are 111001, 100001, 111100, 111101 etc. These are called instances of the schema. The H in the equation stands for hyperplane. A hyperplane is a plane of various dimensions - the dimension of the plane is equal to l, the number of bits in the string. To define this principle further, take the schema: H = * * 0. Since H is a three-dimensional hyperplane it can be graphed like this:



Other traits of a schema include the defined bits and the defining length. The defined bits are the number of bits defined within the schema - in the above example there are 2 defined bits. The defining length is the distance between the two most outer defined bits - in the above example, the defining length is 4.

5.4.2. Sets and Subset

Note that most of the subsets of a length *l* bit string cannot be described as a schema. For a bit string of length-l, there are 2^l possibilities, 2^{2^l} possible substrings, and 3^l (because of the wildcards). Therefore, for a bit string with 4-bits, there are 16 possible bit strings, an amazing 65536 substrings, yet only 81 schemata to represent them in.

5.4.3. Processing

Remembering that any bit string is a member of 2^l different schemata, look at the example Mitchell uses. The length-2 bit string '11' is an instance of **, 1*, *1, and 11. Now, in a population of *n*-strings it is easy to see that you will have anything between 2^l and n x 2^l different schemata.

It can therefore be said, that while the GA *explicitly* evaluates *n* strings during any given generation, it is *implicitly* evaluating a much larger number of schema. Think of it as half of the population being part of the schema '1 * * ... *' and the other half as '0 * * ... *'. Now, after a given generation has been run, you will have an estimate of the average fitness of those schemata (it will be quite a erroneous estimate, since the population actually evaluated was only a tiny portion of the actual schema). The estimates of these averages of obviously not stored explicitly since the schemata themselves are not explicitly evaluated. Where this idea is of help, is in looking at the increase or decrease of a given schema in a population - because it can be described as though the GA were storing these implicitly calculated values.

5.4.4. Compensating for Destructive Effects

Firstly, let p_c be the probability that crossover will be applied to a given string in the population, and assume that an instance of schema H is picked to be the parent. Schema H is said to have "survived" if one of the offstring is still an instance of H. For example, given the schema "1 1 * * * *", and given two strings, 110010 and 010100. Note that the first bit is an instance of the schema, the second is not. Now, if a single-point crossover is applied at the 4th bit, you get the following offspring - 110000 and 010110. The first child is still an instance of the schema, thus it has survived.

Given $S_c(H)$ as being the probability of schema H surviving a single-point crossover, d(H) as the defining length, and l as the length of the bit string you can create the equation:

$$S_C(H) \geq 1 - p_c \times \frac{d(H)}{l-1}$$

A breakdown of this equation is necessary. The d(H)/(l-1) is equal to the fraction of the string that the schema occupies. For example, if we have a defining length of 3 in a string of length 5, then 75% of the string is part of schema H. Now, we multiply that by the probability that they survive, p_c. This gives us an upper-bound that the schema will be destroyed. We want the lower-bound, though, so the upper bound is subtracted from 1 (remember cumulative probabilities equal 1).

Now that crossover has been addressed, we must look at mutation. Let p_m be the probability that a bit will be mutated, and that the function $S_m(H)$ is the probability that the schema H will survive the mutation. In order to survive the mutation, the schema must remain intact, therefore $S_m(H)$ must be equal to the probability that a bit will NOT be mutated ($1-p_m$) raised to the power of the number of defined bits in the schema, or:

$$S_m(H) = (1 - p_m)^{o(H)}$$

where o(H) is the number of defined bits (or *order*) of the schema. From these two calculations you can see that the shorter and lower order the schema, the greater chance it has to survive. By applying these effects with equation 1.1, you get the Schema Theorem first derived by Holland:

$$E(m(H,t+1)) \geq \frac{\hat{u}(H,t)}{\bar{f}(t)} m(H,t)(1 - p_c \frac{d(H)}{l-1})\left((1-p_m)^{o(H)}\right) \qquad \textit{(Eq 2.2)}$$

Hopefully, you now fully understand the mathematics behind genetic algorithms. Note that this equation only deals with the destructive effects of genetic algorithms, in fact crossover is regarded to be one of the chief areas of the GA's power, and mutation is often seen as an insurance policy to prevent a loss of diversity. It is the characteristic of the GA to be able to implicitly evaluate many solutions that makes it so powerful.

5.5. Artificial Intelligent with Genetic Algorithms

Most AI systems are very static. Most of them can usually only solve one given specific problem, since their architecture was designed for whatever that specific problem was in the first place. Thus, if the given problem were somehow to be changed, these systems could have a hard time adapting to them, since the algorithm that would originally arrive to the solution may be either incorrect or less efficient. Genetic algorithms (or GA) were created to combat these problems. They are basically algorithms based on natural biological evolution. The architecture of systems that implement genetic algorithms (or GA) is more able to adapt to a wide range of problems. A GA functions by generating a large set of possible solutions to a given problem. It then evaluates each of those solutions and decides on a "fitness level" for each solution set. These solutions then breed new solutions. The parent solutions that were more "fit" are more likely to reproduce, while those that were less "fit" are more unlikely to do so. In essence, solutions are evolved over time. This way you evolve your search space scope to a point where you can find the solution. Genetic algorithms can be incredibly efficient if programmed correctly.

5.6. General Algorithm for Genetic Algorithms

Genetic algorithms are not too hard to program or understand, since they are biological based. Thinking in terms of real-life evolution may help you understand. Here is the general algorithm for a GA:

5.6.1. Create a Random Initial State

An initial population is created from a random selection of solutions (which are analogous to chromosomes). This is unlike the situation for Symbolic AI systems, where the initial state in a problem is already given instead.

5.6.2. Evaluate Fitness

A value for fitness is assigned to each solution (chromosome) depending on how close it actually is to solving the problem (thus arriving to the answer of the desired problem). (These "solutions" are not to be confused with "answers" to the problem, think of them as possible characteristics that the system would employ in order to reach the answer.)

5.6.3. Reproduce Children Mutate

Those chromosomes with a higher fitness value are more likely to reproduce offspring (which can mutate after reproduction). The offspring is a product of the father and mother, whose composition consists of a combination of genes from them (this process is known as "crossing over".

5.6.4.Next Generation

If the new generation contains a solution that produces an output that is close enough or equal to the desired answer then the problem has been solved. If this is not the case, then the new generation will go through the same process as their parents did. This will continue until a solution is reached.

5.7. Applications of Genetic Algorithms

The possible applications of genetic algorithms are immense. Any problem that has a large search domain could be suitable tackled by GAs. A popular growing field is genetic programming (GP).

5.7.1. Genetic programming

In programming languages such as LISP and Scheme, the mathematical notation is not written in standard notation, but in prefix notation. Some examples of this:

+ 2 1	:	2 + 1
* + 2 1 2	:	2 * (2+1)
* + - 2 1 4 9	:	9 * ((2 - 1) + 4)

```
      +          *          *

     /\         /\         /\

     1  2      +  2       +  9

              /\         /\

              1  2       -  4

                         /\

                         2  1
```

Notice the difference between the left-hand sides to the right? Apart from the order being different, no parenthesis! The prefix method makes life a lot easier for programmers and compilers alike, because order precedence is not an issue. You can build expression trees out of these strings that then can be easily evaluated, for example, here are the trees for the above three expressions.

You can see how expression evaluation is thus a lot easier. What these have to do with GAs? If for example you have numerical data and 'answers', but no expression to conjoin the data with the answers. A genetic algorithm can be used to 'evolve' an expression tree to create a very close fit to the data. By 'splicing' and 'grafting' the trees and evaluating the resulting expression with the data and testing it to the answers, the fitness function can return how close the expression is. The limitations of genetic programming lie in the huge search space the G.A.s have to search for - an infinite number of equations. Therefore, normally before running a G.A. to search for an equation, the user tells the program which operators and numerical ranges to search under. Uses of genetic programming can lie in stock market prediction, advanced mathematics and military.

5.8. Evolving Neural Networks

G.A.s have successfully been used to evolve various aspects of GAs - either the connection weights, the architecture, or the learning function. You can see how GAs are perfect for evolving the weights of a neural network - there are immense number of possibilities that standard learning techniques such as back-propagation would take thousands upon thousands of iterations to converge to. GAs could (given the appropriate direction) evolve working weights within a hundred or so iterations.

Evolving the architecture of neural network is slightly more complicated, and there have been several ways of doing it. For small nets, a simple matrix represents which neuron connection which, and then this matrix is, in turn, converted into the necessary 'genes', and various combinations of these are evolved.

Many would think that a learning function could be evolved via genetic programming. Unfortunately, genetic programming combined with neural networks could be *incredibly* slow, thus impractical. As with many problems, you have to constrain what you are attempting to create. For example, in 1990, David Chalmers attempted to evolve a function as good as the delta rule. He did this by creating a general equation based upon the delta rule with 8 unknowns, which the genetic algorithm then evolved.

5.9. Summary

The development of a design theory for selector combinative G.As. The design and implementation of competent G.A. selector combinative G.As, G.As using selection and recombination that solve hard problems quickly, reliably, and accurately.
Current plans call for the continuation of these efforts in G.A. design theory, competence, efficiency, and application together with new or renewed initiatives to carry the design methodology to genetic programming (G.P.) and to learning classifier systems (L.C.Ss). Additionally, an increased emphasis on solving important problems in computational biology, data mining, and electromagnetic using advanced genetic algorithms, G.P., and L.C.Ss is anticipated.

CHAPTER SIX

APPLICATIONS

6.1. Overview

In this chapter, a little explanation about the applications of previous chapters will be described. Comparison between these applications will be mentioned and some points of view to different scientists applications and my point of view also.

6.2. Applications of Artificial Intelligence in Music

How exactly can artificial intelligence be used in the field of music? A.I. can be used in music in many different ways, it can be used both to compose (create music) and transcribe musical piece Getting computers to compose well is an incredibly hard task. To get a computer to compose a piece requires it to autonomously be able to detect whether a phrases is musical or pure cacophony. There are a few mathematical approaches to determining this, but like anything aesthetic, a certain degree of *gut-feeling* is required to make a piece of significant musical value. Thus, most computer programs that compose require human input to determine whether the music sounds good or not.

One program that does not quite require this is, a program called Variations, developed by Bruce L. Jacob. This program uses genetic algorithms that are evolved the operators liking, *then* it composes and *listen* to the piece to decide whether or not it is good. B.Jacob describes how this genetic "ear" works as:

"...The [ear] module is a collection of chromosomes, each of which acts as a data filter that identifies harmonic combinations as "good" or "bad." Before composition begins, the chromosomes are evolved to reflect the musical tastes of the human operator. First, a set of randomly-generated ear chromosomes are auditioned on how well they filter material. The evaluation mechanism in this process, as in virtually all other genetic music studies, is a human judge. Musical examples are created and passed through the

ear chromosomes, and the human operator assigns weights to chromosomes according to how well they agree with his or her inclinations. Chromosomes with high marks are more likely to reproduce and have their alleles present in the next generation. Successive generations therefore exhibit the best traits of previous generations. Once there is a satisfactory set of filters, the [music creating] process...begins..." [B.Jacob]

A program such as variations creates music autonomously, but what about dynamically? The best example of dynamic music creation is GenJam- or Genetic Jammer. This program by A.I. Biles listens to him play through a device called a *pitch-tracker* that converts the notes from the trumpet into MIDI notes. The program then will improvise using an original set of notes. Biles will then tell GenJam whether the sequence it just played was good or bad. On the next time round, then program will get rid of the tunes that sounded bad, and evolve new parts from the good sounding songs.

6.3. Neural Networks Optimize Enzyme Synthesis

A neural network has been trained to predict the outcome of a chemical reaction controlled by molar ratios, temperature, pressure, amount of enzyme and stirring speed. Used the Brain Maker program to train their neural network to predict the amount of desired product and by-product, which would be formed after 22 hours of reaction time. An excellent correlation between predicted yields and experimental results was found. The neural network saves time and money by predicting the results of chemical reactions so that the most promising conditions can then be verified in the lab, rather than performing a large number of experiments to gain the same information.

Initially 16 experiments were performed to identify the most important parameters controlling the process. The molar ratio between fatty acid and glucoside, reaction temperature, pressure, amount of enzyme, and stirring speed were varied. The synthesis yielded ethyl 6-O-dodecanoyl D-glucopyranoside. This experimental data was used to train the neural network to output the amount of the 6-O monoester and a diester by-product, represented as a percentage of yields.

The neural network had three layers:

1. Five input layer neurons.

2. Four hidden layer neurons.

3. Two output layer neurons.

It was trained using the back propagation algorithm with the sigmoid threshold neuron function. Twelve facts were used to train the network to an accuracy of 96% for the outputs. In only a few minutes, all facts were learned. The trained network was then asked to make four predictions on data it hadn't seen before. The network predictions were compared to experimental observations. Very good correlations were found.

The average deviation between the network and the experiments was 4% (percentage of yield), ranging between 2% and 7% difference. These deviations are within the normal experimental error of synthesis after being tested; the network was put to work evaluating thousands of possible conditions in order to find the most optimum. Using a simple algorithm, a test file was generated containing all of the possible values, totaling 9900 cases.

The computer- generated test file contained values for each parameter which were both within and without of the training value's range. The entire file ran through the network in 7 minutes and the predictions were saved in a file. Using a search function, predictions for specified yields were selected. Only three cases were found to predict more than 88% monoester with a less than 4% formation of the diester. One of these cases was tested in the lab and the results were close to experimental observation. The network had predicted 88.1% monoester and the experiment yielded 86.2%. The network predicted 4.0% diester the experiment yielded 4.8%. Finally, the 9900 predictions were again searched, but this time with additional restrictions more suitable for large-scale chemical processing. Again, the experimental results were very close to the yields predicted by the network

6.4. Expert Systems in HIPAA and Electronic Medical Records

HIPAA requires that the security and privacy policies of the provider organization (your practice) be recorded and audited for compliance. The difficulty with achieving this compliance can be simplified by utilizing an electronic medical record (EMR) in the practice.

An EMR may assist the practice in achieving compliance with HIPAA's security and privacy requirements in the following ways:

1. **Role-based authentication:** Different providers and/or office personnel can be granted/denied access to certain fields, data or patients as well as complete blockage unless it is necessary for the person to complete their job. Multiple access levels can be created to accommodate the many roles are served by your office personnel.

2. **Access is recorded and audited:** Anyone accessing a patient's EMR will leave an audit trail for review by supervisory personnel. Any change to a patient's record will record the date, time and person who made the adjustment.

3. Protected health information can be encrypted or de-identified. Sorting through your patient documentation via an EMR is much easier than pulling the numerous files that paper records creates. By randomly assigning record numbers or running filters, your EMR will allow you to de-identify information without destroying the record.

4. **Data integrity:** The information stored within your EMR will be protected from unauthorized access. Security levels established within your EMR will only allow access to those who should be reviewing and amending the data.

5. **Archive of data:** Because your patient information is in electronic form, you are better able to back up and archive the data which not only protects the data, but saves valuable office space and money.

6.4.1. Problems of Expert Systems

Expert systems cannot be applied in all areas, in some areas it may make an errors in it and here some of the problems of an expert systems

- Expert systems are likely to succeed in some, but not all, problem domains. Often these are very narrowly defined domains.

- It may be desirable to develop hybrid systems, in an attempt to merge human and machine expertise.

- Key (sought-after) features in future generation ES's are likely to be:

1. Efficient and user-friendly methods of knowledge acquisition.

2. Expanded explanation-generating capability (accountability), especially in ES's used for diagnosis.

3. Successful strategies for incorporating uncertainty in the representation.

6.5. Comparison of neural networks, expert systems and Artificial Intelligence

Artificial neural networks offer a completely different approach to problem solving and they are sometimes called the sixth generation of computing. They try to provide a tool that both programs itself and learns on its own. Neural networks are structured to provide the capability to solve problems without the benefits of an expert and without the need of programming. They can seek patterns in data that no one knows are there. A comparison of artificial intelligences expert systems and neural networks is contained in. Expert systems have enjoyed significant successes. However, artificial intelligence has encountered problems in areas such as vision, continuous speech recognition and synthesis, and machine learning. Artificial intelligence also is hostage to the speed of the processor that it runs on. Its is restricted to the theoretical limit of a single processors. Artificial intelligence is also burdened by the fact that experts don't always speak in rules.

6.6. Applications of Genetic Algorithms in Scheduling

Genetic Algorithm is used for inspection and repair of oil tanks and pipelines. The implementation is built on Peter Ross' PGA testbed and the data is taken from the Expert Systems for the Inspection of Tanks and Pipelins SITA and SIGO. The fitness function evaluates the constraints: level of production, condition and location of installations, type of products, human resources, the dates and costs of inspection and repairs. A good inspection schedule for oil installations is constructed. A good schedule ensures that repair times are kept to a minimum and faults are found before they become too serious. An automatic way of assigning maintenance activities to inspectors is devised in such a way as to minimize the loss in capacity, while keeping within resource constraints.

The schedule is evaluated taking into account the following priorities:

1. A tank which requires urgent maintenance is checked early in the schedule (very good).

2. A tank or pipeline requiring a periodic maintenance or inspection is included in the schedule (good), given higher priority to the first case.

3. Because of several tanks in one location being out of service simultaneously, the capacity of that location for a certain time drops significantly (very bad).

4. Some inspectors have more work to do than the others in the same area (bad).

The application distributes the repairs such that the available capacity is always larger than the required minimum, then the production is not affected. Moreover, the assignment of activities is appropriate, it reduces the cost of unbalanced distribution. A robust schedule of activities is obtained.

Genetic Algorithm uses the feedback from the evaluation process to select the fitter designs, generating new designs through the recombination of parts of the selected designs. Eventually, a population of high performance designs are resulted.

6.7. Benefits of genetic algorithms in specific areas

Traditional methods of search and optimization are too slow in finding a solution in a very complex search space, even implemented in supercomputers. Genetic Algorithm is a robust search method requiring little information to search effectively in a large or poorly-understood search space. In particular a genetic search progress through a population of points in contrast to the single point of focus of most search algorithms. Moreover, it is useful in the very tricky area of nonlinear problems. Its intrinsic parallelism (in evaluation functions, selections and so on) allows the uses of distributed processing machines.

6.8. Summary

In this, chapter simple and various applications are mentioned about each chapter; a comparison between different types of applications, some comments to different scientists and the benefits of each application in its own area.

CONCLUSION

In this project, systems are continuously being mentioned; but what are systems? Systems can be defined as a part of the universe, containing parts that have stronger correlations with each other than with the rest of the universe. While looking for this definition, we found that there are really two types of causes. One type is related to the changes caused by an intelligent system, and the other type is related to those changes occurring in nature, where one occurrence only has a correlation with another.

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.

Finally, it is useful to learn from the artificial IS and thereby increase our own intelligence.

It is amazing to observe how a small thing such as a rigorous definition of an intelligent system can have ramifications on our understanding of societies, ethics, and philosophical outlook.

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, once artificial I.S.'s are working well, they will certainly think much faster that humans can (their internal clock has millions of cycles per second) and they will be more exact in thinking. We humans often confuse concepts or have very inexact concepts. Once A.I.S.'s in computers are in use for many years, their concepts should be very precise and very detailed. We can easily copy this structure of concepts and response rules into all new A.I.S.'s. Therefore, we see that in the end computers will think much better and much faster than humans ever can. Still they are machines, we give them their objectives, and they do not give us our objectives.

In chapter three, "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 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, "Genetic Algorithms" are not too hard to program or understand, since they are biological based. Thinking in terms of real-life evolution may help you to understand and there is a general algorithm for the genetic algorithms.

In chapter six, "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.

REFERNCES

[1] Assoc. Prof. Dr *Adnan Khashman,* Lectures Note "Introduction to Neural Networks" fall semester [2001-2002]

[2] Determining Economic Equilibria using Genetic Algorithms, by Meliha Chughtai, in September 1995. Unpublished Thesis. [Chughtai 1995]

[3] "Intelligent systems and their societies" Edited by Walter Fritz(www.anice.net.ar)

[4] "Computers for Artificial Intelligence Processing" Edited by Benjamin Wah & C.V. Ramonmoorth

[5] "Introduction to Artificial Intelligence" Edited by K.Warwick.

[6] Internet: www.google.com.

[7] Internet: www.pioneer.com.

[8] Internet: www.arabia.com.

[9] Internet: www.altavista.com.

[10] Internet: www.askme.com.

[11] internet: www.hotbot.com.

[12] Internet: World Wide Web.