NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

INDUSTRIAL APPLICATIONS OF NEURAL NETWORKS

Graduation Project

COM – 400

Student: Hafez Abu Jabal

Supervisor: Assoc. Prof. Dr. Adnan Khashman

Nicosia - 2003

ABSTRACT

Throughout the series chapters of the project, which is including and also will be illustrated the definitions, the structure and many applications and to be much more specific applications in industry, applying one of the most important and essential technology among the other technologies which is the Artificial Neural Networks let it be said the Giant of the technologies.

This project describes the benefits which are given by artificial neural networks in the area of industry, process monitoring and control. Descriptions of the major activities undertaken in this project, which included the application of neural networks for the application of neural networks to the paper-making industry, Quality Inspection of Wood Surfaces.

The project also describes some of the practical difficulties that were experienced while applying neural networks within industry and lists the important lessons that were learned through the completion of this project. The main conclusion from the project was that neural networks are capable of improving industrial process monitoring and control systems. However the level of improvement must be analysed on a problem specific basis and in many applications the use of neural networks may not be justified.

No doubt that Artificial Neural Networks have played major and significant rule in our life, it has been found in lots of field in real life applications, such as in medicine, forecasting, recognitions and in industry. Though, no one can deny that gorgeous rule to the Artificial Neural Networks in our life.

TABLE OF CONTENTS

INTRODUCTION

Artifcial Neural Networks (ANNs) have been studied almost from the beginning of the computer era. In the beginning the NN research was strongly motivated by biological considerations and the developed NN models were too weak to solve complex information processing tasks typically found in many industrial applications. New innovations in 1980s lead to the emerge of more powerful NN models, and many successful case studies aimed at demonstrating NN concepts and benefits to various categories in a number of industrial application sectors caused the industry to look the NNs as a serious tools. Nowadays NNs have popular appeal, and a wide application sector within the industry.

Chapter One: in this chapter it will be tried to focus on the verious definitions of the Artificial Neural Networks, what are the Artificial Neural Networks, a brief history about A.N.Ns. since the first trials to recent days, moreover, what makes A.N.Ns. differ from traditional computing and expert systems, as whats the binefits of using A.N.Ns. in real life applications, is there any limits to the technology of Artificial Neural Networks and the future of it and what applications could be involved in is also will be retrieved.

Chapter Two: within this chapter the archeticture of Neural Networks and some of its topologies is going to be shown and also discussed; the way that A.N.Ns. is behaiving within the teaching the N.N. session will also be illustrated, till the next development.

Chapter Three: this chapter wil be concern about spesific real life applications of N.Ns. these applications will be in industrial sectors, so some of these applications are: the quality inspection of wood surfarces, NASA developed a new software which going to take the plane or the aircraft to a safe land if something fault happend which may cause damage "disaster" which threats people's life on its boards.

Chapter Four: will be about one topic or one industrial application, as within chapter three N.Ns. was involved in safety within the indutrial applications, so here Industrial use of safety-related artificial neural networks with many applications will be retrieved to illustrate it and make it clear how important N.Ns. are.

The main objectives of this project were to illustrate the major rule of A.N.Ns. in man's life and what have it improved and still improving and solving especially in industrial applications.

CHAPTER ONE
INTRODUCTION TO NEURAL NETWORKS

1.1 Overview

Within this chapter, many characteristics of Artificial Neural Networks will be discussed, first of all, a hint will be given about Artificial Neural Networks A.N.Ns. and what is A.N.N. Historical background to A.N.Ns. will also be included, more over what makes Neural Network differ from Traditional Computing is also has been taken in consideration. Also in this chapter, where do A.N.N. exists and why? will be illustrated as well, is there any limits for A.N.Ns. and its future will be reviewed, some of the advantages and disadvantages will be listed.

1.2 Artificial Neural Networks

Artificial neural networks can be seen as highly parallel dynamical systems consisting of multiple simple units that can perform transformations by means of their state response to their input information. How the transformation is carried out depends on the N.N. model and its way of learning the transformation. Neural networks learn by example. In a typical scenario, a neural network is presented iteratively with a set of samples, known as the training set, from which the network can learn the values of its internal parameters.

Over the last few years many different N.N. models have been proposed, for a broad range of applications from remote sensing, to speech recognition to prediction of financial indices. Some components of these models date from much earlier such as the perceptron element (Rosenblatt) [1]. Foremost amongst these models for supervised classification tasks, are the multi-layer perceptron or multi-layer feed-forward neural network and the learning vector quantisation (LVQ) network. The multi-layer perceptron came into favour when Rumelhart *et al* [2]. "re-discovered" the gradient back-propagation algorithm for training layered networks of perceptron elements (although their method can in fact be traced earlier to Werbos) [3]. The LVQ method was developed by Kohonen [4] who also developed the popular unsupervised classification technique known as the self-organising map or topological map neural networks. Together these three neural network techniques, multi-layer perceptron learning vector quantisation and topological map account for the vast majority of reported applications of neural networks in remote sensing; and of these three, the multi-layer perceptron far outweighs the others in terms of number of experimental tests.

A lot of effort has been done in laying the theoretical foundations of N.N.s and the links between statistical and neural methodologies. Many N.N.s models, such as adaline, multi-layer perceptron, or PCA networks are similar or identical to more conventional statistical techniques such as generalised linear models, polynomial regression, non-parametric regression and discriminant analysis, principal components, and cluster analysis. There are also a few useful N.N. models, such as counterpropagation, learning vector quantisation, and self-organising maps, that have no precise statistical equivalent but are useful in data analysis.

1.3 What are Artificial Neural Networks

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

Artificial Neural Network is a system loosely modeled on the human brain. The field goes by many names, such as connectionism, parallel distributed processing, neuro-computing, natural intelligent systems, machine learning algorithms, and artificial neural networks. It is an attempt to simulate within specialized hardware or sophisticated software, the multiple layers of simple processing elements called neurons. Each neuron is linked to certain of its neighbors with varying coefficients of connectivity that represent the strengths of these connections. Learning is accomplished by adjusting these strengths to cause the overall network to output appropriate results. According to the DARPA Neural Network Study (1988, AFCEA International Press, p. 60) [5].

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

According to Haykin, S [6].

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

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

Neural computing is the study of networks of adaptable nodes which, through a process of learning from task examples, store experiential knowledge and make it available for use.

1.4 Historical Background of Neural Networks

Neural network simulations appear to be a recent development. However, this field was established before the advent of computers, and has survived at least one major setback and several eras.

Many importand advances have been boosted by the use of inexpensive computer emulations. Following an initial period of enthusiasm, the field survived a period of frustration and disrepute. During this period when funding and professional support was minimal, important advances were made by relatively few reserchers. These pioneers were able to develop convincing technology, which surpassed the limitations identified by Minsky and Papert. Minsky and Papert, published a book (in 1969) in which they summed up a general feeling of frustration (against neural networks) [7] among researchers, and was thus accepted by most without further analysis. Currently, the neural network field enjoys a resurgence of interest and a corresponding increase in funding.

The history of neural networks that was described above can be divided into several periods:

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

functions such as "a or b" and "a and b". Another attempt was by using computer simulations. Two groups (Farley and Clark, 1954; Rochester, Holland, Haibit and Duda, 1956). The first group (IBM reserchers) maintained closed contact with neuroscientists at McGill University. So whenever their models did not work, they consulted the neuroscientists. This interaction established a multidiscilinary trend which continues to the present day [8].

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

Another system was the ADALINE (*ADAptive LInear Element*) which was developed in 1960 by Widrow and Hoff (of Stanford University). The ADALINE was an analogue electronic device made from simple components. The method used for learning was different to that of the Perceptron, it employed the Least-Mean-Squares (LMS) learning rule [9].

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

4. Innovation: Although public interest and available funding were minimal, several researchers continued working to develop neuromorphically based computaional methods for problems such as pattern recognition.

During this period several paradigms were generated which modern work continues to enhance.Grossberg's (Steve Grossberg and Gail Carpenter in 1988) [10] influence founded a school of thought which explores resonating algorithms. They developed the ART (Adaptive Resonance Theory) networks based on biologically plausible models. Anderson and Kohonen developed associative techniques independent of each other. Klopf (A. Henry Klopf) in 1972 [11], developed a basis for learning in artificial neurons based on a biological principle for neuronal learning called *heterostasis.* Werbos (Paul Werbos 1974) [12] developed and used the *back-propagation* learning method, however several years passed before this approach was popularized. Back-propagation nets are probably the most well known and widely applied of the neural networks today. In essence, the back-propagation net. is a Perceptron with multiple layers, a different thershold function in the artificial neuron, and a more robust and capable learning rule.

Amari (A. Shun-Ichi 1967) was involved with theoretical developments: he published a paper which established a mathematical theory for a learning basis (error-correction method) dealing with adaptive patern classification. While Fukushima (F. Kunihiko) [13] developed a step wise trained multilayered neural network for interpretation of handwritten characters. The original network was published in 1975 and was called the *Cognitron.*

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

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

1.5 Analogy to the Brain

The exact workings of the human brain are still a mystery. Yet, some aspects of this amazing processor are known. In particular, the most basic element of the human brain is a specific type of cell which, unlike the rest of the body, doesn't appear to regenerate. Because this type of cell is the only part of the body that isn't slowly replaced, it is assumed that these cells are what provides us with our abilities to remember, think, and apply previous experiences to our every action. These cells, all 100 billion of them, are known as neurons. Each of these neurons can connect with up to 200,000 other neurons, although 1,000 to 10,000 is typical.

The power of the human mind comes from the sheer numbers of these basic components and the multiple connections between them. It also comes from genetic programming and learning.

The individual neurons are complicated. They have a myriad of parts, sub-systems, and control mechanisms. They convey information via a host of electrochemical pathways. There are over one hundred different classes of neurons, depending on the classification method used. Together these neurons and their connections form a process which is not binary, not stable, and not synchronous. In short, it is nothing like the currently available electronic computers, or even artificial neural networks.

These artificial neural networks try to replicate only the most basic elements of this complicated, versatile, and powerful organism. They do it in a primitive way. But for the software engineer who is trying to solve problems, neural computing was never about replicating human brains. It is about machines and a new way to solve problems.

1.5.1 The Perceptron

The perceptron is a mathematical model of a biological neuron figure 1.1 and figure 1.2. While in actual neurons the dendrite receives electrical signals from the axons of other neurons, in the perceptron these electrical signals are represented as numerical values. At the synapses between the dendrite and axons, electrical signals are modulated in various amounts. This is also modeled in the perceptron by multiplying each input value by a value called the weight. An actual neuron fires an output signal only when the total strength of the input signals exceed a certain threshold. We model this phenomenon in a perceptron by calculating the weighted sum of the inputs to represent the total strength of the input signals, and applying a step function on the sum to determine its output. As in biological neural networks, this output is fed to other perceptrons.

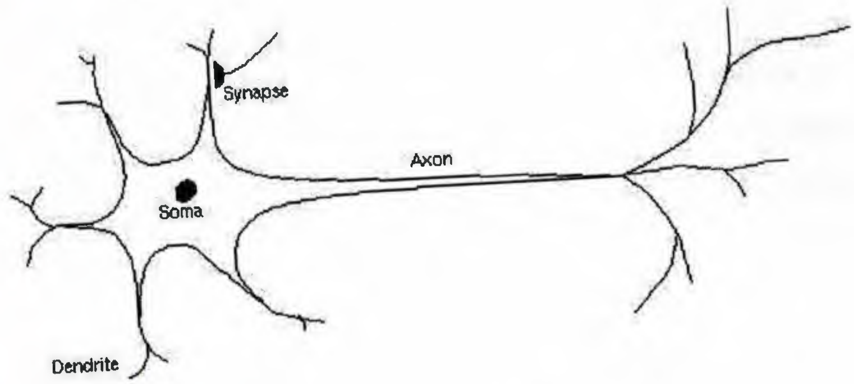

Figure 1.1 A biological neuron

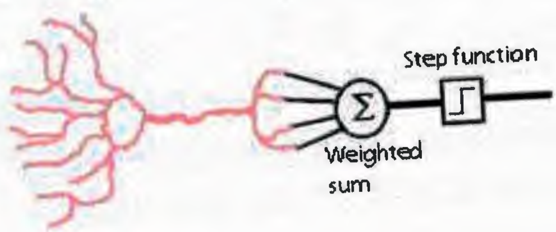

Figure 1.2 An artificial neuron (perceptron)

There are a number of terminologies commonly used for describing neural networks. They are listed in the table below (table 1.1):

Table 1.1 number of terminologies commonly used for describing neural networks

The input vector	All the input values of each perceptron are collectively called the input vector of that perceptron.
The weight vector	Similarly, all the weight values of each perceptron are collectively called the weight vector of that perceptron.

1.6 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 1.2 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. 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, that 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.

Table 1.2 Comparison of Computing Approaches.

CHARACTERISTICS	TRADITIONAL COMPUTING (including Expert Systems)	ARTIFICIAL NEURAL NETWORKS
Processing style Functions	Sequential Logically (left brained) via Rules Concepts Calculations	Parallel Gestault (right brained) via Images Pictures Controls
Learning Method Applications	by rules (didactically) Accounting word processing math inventory digital communications	by example (Socratically) Sensor processing speech recognition pattern recognition text recognition

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 intelligence's expert systems and neural networks is contained in Table 1.3.

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. Ultimately, it is restricted to the theoretical limit of a single processor. Artificial intelligence is also burdened by the fact that experts don't always speak in rules.

Table 1.3 Comparisons of Expert Systems and Neural Networks.

Characteristics	Von Neumann Architecture Used for Expert Systems	Artificial Neural Networks
Processors	VLSI (traditional processors)	Artificial Neural Networks; variety of technologies; hardware development is on going
Processing Approach	Separate	The same
Processing Approach	Processes problem rule at a one time; sequential	Multiple, simultaneously
Connections	Externally programmable	Dynamically self programming
Self learning	Only algorithmic parameters modified	Continuously adaptable
Fault tolerance	None without special processors	Significant in the very nature of the interconnected neurons
Neurobiology in design	None	Moderate
Programming	Through a rule based complicated	Self-programming; but network must be set up properly
Ability to be fast	Requires big processors	Requires multiple custom-built chips

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 implementor 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 activization 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 the traditional computing environments. For example, speech is something that all people can easily parse and understand. 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.

This 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.

1.7 Why Use a Neural Network

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

Other advantages include:

1. Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.
2. Self-Organisation: An ANN can create its own organisation or representation of the information it receives during learning time.
3. Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manifactured which take advantage of this capability.
4. Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilites may be retained even with major network damage.

1.8 Neural Networks in Practice

Neural networks are performing successfully where other methods do not, recognizing and matching complicated, vague, or incomplete patterns. Neural networks have been applied in solving a wide variety of problems.

The most common use for neural networks is to project what will most likely happen. There are many areas where prediction can help in setting priorities. For example, the emergency room at a hospital can be a hectic place, to know who needs the most critical help can enable a more successful operation. Basically, all organizations must establish priorities, which govern the allocation of their resources. Neural networks have been used as a mechanism of knowledge acquisition for expert system in stock market forecasting with astonishingly accurate results. Neural networks have also been used for bankruptcy prediction for credit card institutions.

Although one may apply neural network systems for interpretation, prediction, diagnosis, planing, monitoring, debugging, repair, instruction, and control, the most successful applications of neural networks are in categorization and pattern recognition. Such a system classifies the object under investigation (e.g. an illness, a pattern, a picture, a chemical compound, a word, the financial profile of a customer) as one of numerous possible categories that, in return, may trigger the recommendation of an action (such as a treatment plan or a financial plan.

One of the best-known applications is the bomb detector installed in some U.S. airports. This device called SNOOPE, determine the presence of certain compounds from the chemical configurations of their components.

Given this description of neural networks and how they work, what real world applications are they suited for? Neural networks have broad applicability to real world business problems. In fact, they have already been successfully applied in many industries.

Since neural networks are best at identifying patterns or trends in data, they are well suited for prediction or forecasting needs including:

- sales forecasting
- industrial process control
- customer research
- data validation
- risk management
- target marketing

But to give some more specific examples; ANN are also used in the following specific paradigms: recognition of speakers in communications; diagnosis of hepatitis; recovery of telecommunications from faulty software; interpretation of multimeaning Chinese words; undersea mine detection; texture analysis; three-dimensional object recognition; handwritten word recognition; and facial recognition.

1.9 Neural Networks Versus Conventional Computers

Neural networks take a different approach to problem solving than that of conventional computers. Conventional computers use an algorithmic approach i.e. the computer follows a set of instructions in order to solve a problem. Unless the specific steps that the computer needs to follow are known the computer cannot solve the problem. That restricts the problem solving capability of conventional computers to problems that we already understand and know how to solve. But computers would be so much more useful if they could do things that we don't exactly know how to do.

Neural networks process information in a similar way the human brain does. The network is composed of a large number of highly interconnected processing elements(neurones) working in parallel to solve a specific problem. Neural networks learn by example. They cannot be programmed to perform a specific task. The examples must be selected carefully otherwise useful time is wasted or even worse the network might be functioning incorrectly. The disadvantage is that because the network finds out how to solve the problem by itself, its operation can be unpredictable.

On the other hand, conventional computers use a cognitive approach to problem solving; the way the problem is to solve must be known and stated in small unambiguous instructions. These instructions are then converted to a high level language program and then into machine code that the computer can understand. These machines are totally predictable; if anything goes wrong is due to a software or hardware fault.

Neural networks and conventional algorithmic computers are not in competition but complement each other. There are tasks are more suited to an algorithmic approach like arithmetic operations and tasks that are more suited to neural networks. Even more, a large number of tasks require systems that use a combination of the two approaches (normally a conventional computer is used to supervise the neural network) in order to perform at maximum efficiency.

1.10 Are There Any Limits to Neural Networks

The major issues of concern today are the scalability problem, testing, verification, and integration of neural network systems into the modern environment. Neural network programs sometimes become unstable when applied to larger problems. The defence, nuclear and space industries are concerned about the issue of testing and verification. The mathematical theories used to guarantee the performance of an applied neural network are still under development. The solution for the time being may be to train and test these intelligent systems much as we do for humans. Also there are some more practical problems like:

- the operational problem encountered when attempting to simulate the parallelism of neural networks. Since the majority of neural networks are simulated on sequential machines, giving rise to a very rapid increase in processing time requirements as size of the problem expands.

Solution: implement neural networks directly in hardware, but these need a lot of development still.

- instability to explain any results that they obtain. Networks function as "black boxes" whose rules of operation are completely unknown.

1.11 The Future

Because gazing into the future is somewhat like gazing into a crystal ball, so it is better to quote some "predictions". Each prediction rests on some sort of evidence or established trend which, with extrapolation, clearly takes us into a new realm.

Prediction 1: Neural Networks will fascinate user-specific systems for education, information processing, and entertainment. "Alternative ralities", produced by comprehensive environments, are attractive in terms of their potential for systems control, education, and entertainment. This is not just a far-out research trend, but is something which is becoming an increasing part of our daily existence, as witnessed by the growing interest in comprehensive "entertainment centers" in each home. This "programming" would require feedback from the user in order to be effective but simple and "passive" sensors (e.g fingertip sensors, gloves, or wristbands to sense pulse, blood pressure, skin ionisation, and so on), could provide effective feedback into a neural control system. This could be achieved, for example, with sensors that would detect pulse, blood pressure, skin ionisation, and other variables which the system could learn to correlate with a person's response state.

Prediction 2: Neural networks, integrated with other artificial intelligence technologies, methods for direct culture of nervous tissue, and other exotic technologies such as genetic engineering, will allow us to develop radical and exotic life-forms whether man, machine, or hybrid.

Prediction 3: Neural networks will allow us to explore new realms of human capabillity realms previously available only with extensive training and personal discipline. So a specific state of consiously induced neurophysiologically observable awareness is necessary in order to facilitate a man machine system interface.

1.12 Advantages and Disadvantages of Neural Networks

Asides from people's fear of artificial neural networks, A.N.N. has several advantages and disadvantages. Because A.N.N. is similar to B.N.N, if parts of the network are damaged, it can still carry on its works. Another advantage is its ability to learn from limited sets of examples. For instance, a handwriting recognition program can recognize handwriting even though it has only been trained using several people's handwriting. However, unlike traditional programs, if parts of the program are damaged, it could no longer function. Furthermore, the same neural network can be used for several programs without any modification. An example of that would be Optical Character Recognition (O.C.R.) programs. The neural network used in English (O.C.R.) programs can be used in Chinese versions because the network is designed to learn patterns. By retraining the network, and changing the database, the program for the network does not need to be modified and can still do its tasks.

The speed of the A.N.N. can be both its advantage and disadvantage. Depending on the level of AI required, a network with a larger input, hidden, and output layers may be required. If the computer is not fast enough to process the information, a tremendous amount of time may be required to process a simple question. The complexity of the network is considered to be its disadvantage because you don't know whether the network has "cheated" or not. Because a neural network can memorize and recognize patterns, it is almost impossible to find out how the network comes up with its answers. This is also known as a black box model. For example, you can provide a neural network with several pictures of a person and ask it to recognize him/her. However, there's no way to guarantee the network will recognize the person because it is possible that the network memorized the photos and, when new pictures are given, it cannot tell who the person is. Further more, it is also possible that the network recognizes the background instead of the person. Hence using neural network for any kind of recognition could be risky. When you ask a neural network to recognize a tank in the forest by providing pictures of forest with tanks and pictures of forests without tanks, it may simply recognize the weather condition taken for the two different categories. Due to the problem just described, it is essential to test the network after its training by introducing it to other inputs that the network has never experienced before.

1.13 Summary

Through this chapter what is Artificial Neural Network, has been discribed, the history of A.N.Ns. and what makes the technology of Artificial Neual Networks different from tradituional computing, have also been listed within this chapter. Also the Artificial Neural Networks in the future and what kind of fields it may go through to solve some problem that could serve the humanity.

Some advantages and disadvantages of the A.N.N. also have been covered through this chapter.

CHAPTER TWO

ARCHITECTURE OF NEURAL NETWORKS

2.1 Overview

Within this chapter neural network architecture is going to be listed and illustrated. Starting from the characteristics of A.N.Ns. Which contain the structure, processing elements to the training session of Artificial Neural Networks up to where A.N.Ns. will be used is selected to be reviewed in here. The concern here is to have a Neural Network architecture which fits the objectives of the task at hand - measured by the generalisation ability of the neural network model as opposed to being tuned only on the training sample available. This chapter describes the strategy acquired the necessary data to train the network. Further it involves the selection of learning rules, transfer functions, summation functions, and how to connect the neurons within the network.

2.2 Characteristics of Neural Networks

The ANN can be characterized by the following features:

- The *network architecture*, which specifies the arrangement of neural interconnections and type of processing elements PEs. The network structure is discussed below.
- The *network processing* specifies how the PEs for a given set of weights calculate the output vector y for an input vector x. It depends on the ANN architecture and the type of transfer function of the PEs.
- The *training algorithm* specifies how the ANN adapts its weights for all M input vectors x, called training vectors, from a set of given vectors, the training set.

2.2.1 Network Structure

Neural computing requires a number of elementary processing elements (PE), also called neurons, to be connected together into an ANN. In other words, the ANN consists of many PEs joined together. Interconnection of PEs is defined by network topology.

A typical network architecture is shown at the figure below:

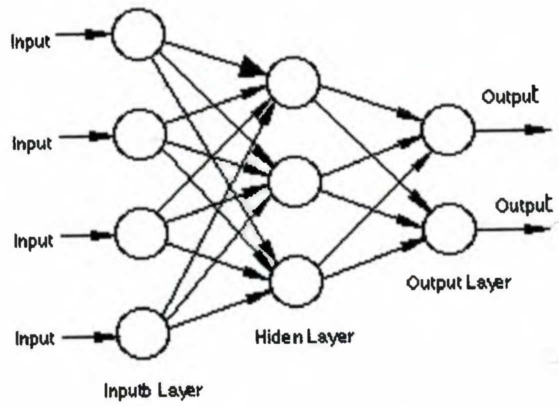

Figure 2.1 a Typical Neural Network

Processing elements are usually organized into groups called layers or slabs. A typical network consists of a sequence of layers or slabs full or random connections between succesive layers. The data then passes through the network to the output layer to provide the solution or answer. There are typically two layers with connections to the outside world:

- An input buffer where data is presented to the network,
- and output buffer which holds the response of the network to a given input.

Layers distinct from the input and output buffers are called hidden layers.

2.2.2 A Processing Element

In the previous chapter, a simplified model of a biological neuron called Threshold Logic Unit (TLU) was introduced. In this chapter, we define some new features to enhance its capabilities.

In the ANN literature, the unit analoguous to the biological neuron is refered to as a processing element. A processing element (PE) has many input paths (analoguous to biological neuron's dendrites) and combines, usually by a simple summation, the values of these input paths. A view of the processing element is shown below.

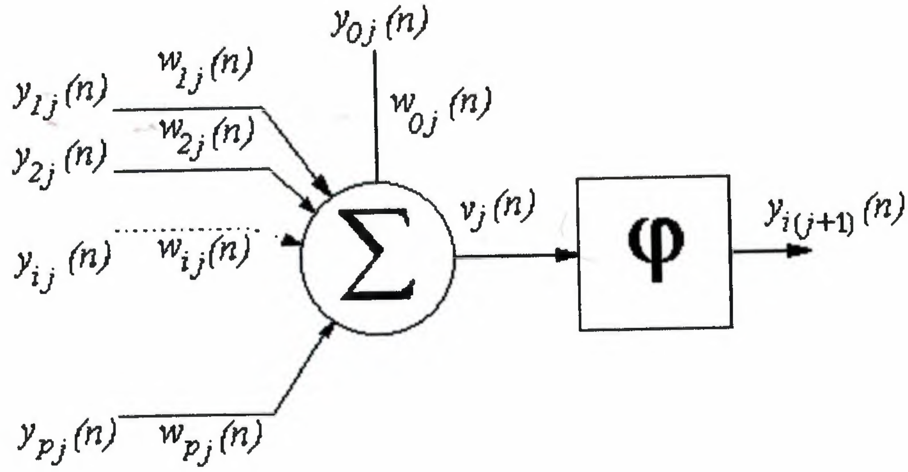

Figure 2.2 a Processing Element

The result is an internal activity level v for the PE. The combined input v is then modified by a transfer function (also refered to as activation, or gain function). This transfer function can be a threshold function which only passes information if the combined activity level reaches a certain level, or it can be a continuous function of the combined input. The output value of the transfer function is generally passed directly to the output path of the PE (except to some network paradigms).

$$v_j^{(n)} = \sum_{i=0}^{p} y_{ij}^{(n)} w_{ij}^{(n)}$$

$$y_{i(j+1)}^{(n)} = \varphi(v_j^{(n)})$$

where:

v_j Is the internal activity level of the PE in the jth layer,

y_{ij} Is the ith input of the PE in the jth layer,

w_{ij} Is the weight of the ith input of the PE in the jth layer,

$\boldsymbol{P}$ Is the number of inputs to PE

$n$denotes the nth input vector presented to the PE.

Examples of the transfer functions are shown below.

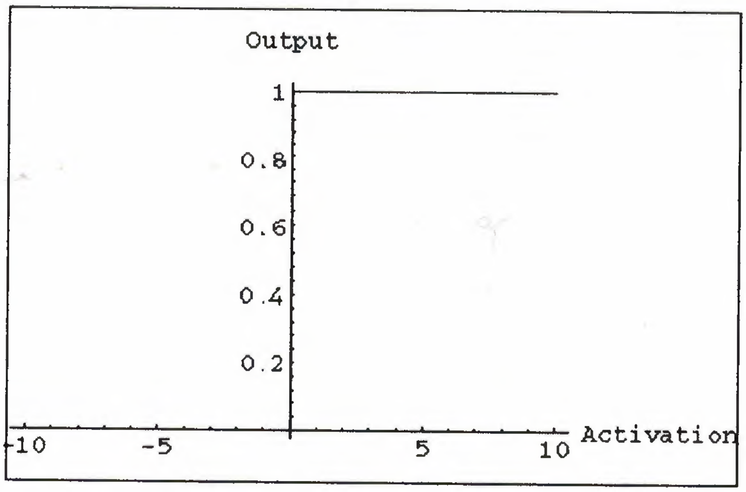

Figure 2.3 Threshold transfer function with different parameters

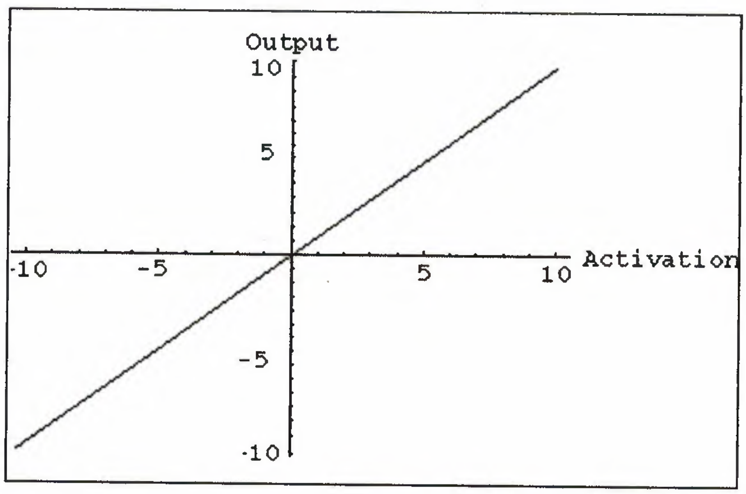

Figure 2.4 Linear transfer function

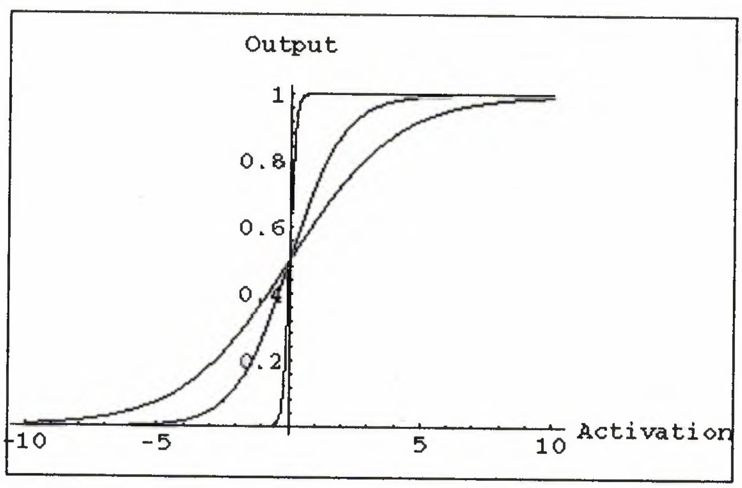

Figure 2.5 Sigmoid transfer function with different parameters

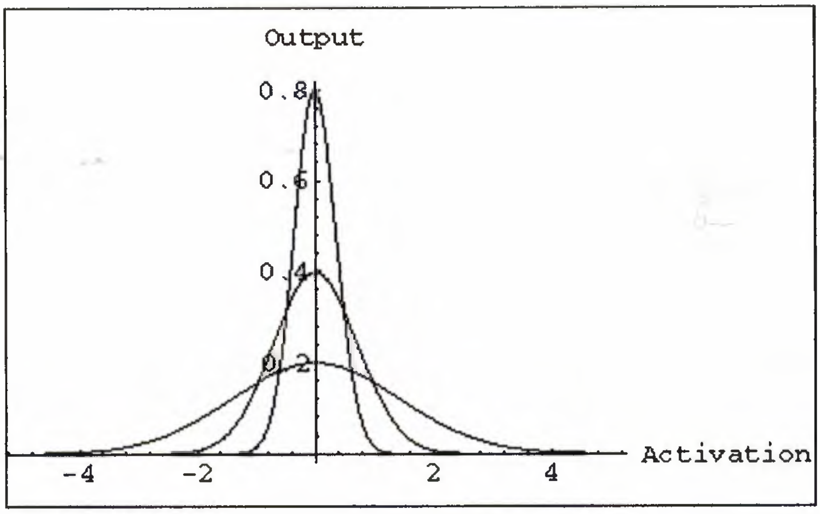

Figure 2.6 Gaussian transfer function with different parameters

The output path of a processing element can be connected to input paths of other PE through connection weights which correspond to the synaptic strength of neural connections. Since each connection has a corresponding weight, the signals on the input lines to a processing element are modified by these weight prior to being summed. Thus the summation function is a weighted summation.

2.3 Training an Artificial Neural Network

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.

2.3.1 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 nonsignificant 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 overtrained. 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.

2.3.2 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 adaption.

At the present time, unsupervised learning is not well understood. This adaption 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, an electrical engineer at the Helsinki University of Technology [14]. 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, feedforward, 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.

2.3.3 Training Rules

Many learning laws are in common use. Most of these laws are some sort of variation of the best known and oldest learning law, Hebb's Rule. Research into different learning functions continues as new ideas routinely show up in trade publications. Some researchers have the modeling of biological learning as their main objective. Others are experimenting with adaptations of their perceptions of how nature handles learning. Either way, man's understanding of how neural processing actually works is very limited. Learning is certainly more complex than the simplifications represented by the learning laws currently developed. A few of the major laws are presented as examples.

Hebb's Rule: The first, and undoubtedly the best known, learning rule was introduced by Donald Hebb. The description appeared in his book *The Organization of Behavior* in 1949. His basic rule is: If a neuron receives an input from another neuron, and if both are highly active (mathematically have the same sign), the weight between the neurons should be strengthened.

Hopfield Law: It is similar to Hebb's rule with the exception that it specifies the magnitude of the strengthening or weakening. It states, "if the desired output and the input are both active or both inactive, increment the connection weight by the learning rate, otherwise decrement the weight by the learning rate."

The Delta Rule: This rule is a further variation of Hebb's Rule. It is one of the most commonly used. This rule is based on the simple idea of continuously modifying the strengths of the input connections to reduce the difference (the delta) between the desired output value and the actual output of a processing element. This rule changes the synaptic weights in the way that minimizes the mean squared error of the network. This rule is also referred to as the Widrow-Hoff Learning Rule and the Least Mean Square (LMS) Learning Rule.

The way that the Delta Rule works is that the delta error in the output layer is transformed by the derivative of the transfer function and is then used in the previous neural layer to adjust input connection weights. In other words, this error is back-propagated into previous layers one layer at a time. The process of back-propagating the network errors continues until the first layer is reached. The network type called Feedforward, Back-propagation derives its name from this method of computing the error term.

When using the delta rule, it is important to ensure that the input data set is well randomized. Well ordered or structured presentation of the training set can lead to a network which can not converge to the desired accuracy. If that happens, then the network is incapable of learning the problem.

The Gradient Descent Rule: This rule is similar to the Delta Rule in that the derivative of the transfer function is still used to modify the delta error before it is applied to the connection weights. Here, however, an additional proportional constant tied to the learning rate is appended to the final modifying factor acting upon the weight. This rule is commonly used, even though it converges to a point of stability very slowly.

It has been shown that different learning rates for different layers of a network help the learning process converge faster. In these tests, the learning rates for those layers close to the output were set lower than those layers near the input. This is especially important for applications where the input data is not derived from a strong underlying model.

Kohonen's Learning Law: This procedure, developed by Teuvo Kohonen, was inspired by learning in biological systems. In this procedure, the processing elements compete for the opportunity to learn, or update their weights. The processing element with the largest output is declared the winner and has the capability of inhibiting its competitors as well as exciting its neighbors. Only the winner is permitted an output, and only the winner plus its neighbors are allowed to adjust their connection weights

2.4 Classification of Neural Networks

Neural Network models can be classified in a number of ways. Using the network architecture as basis, there are three major types of neural networks:

2.4.1 Recurrent Networks

The units are usually laid out in a two-dimensional array and are regularly connected. Typically, each unit sends its output to every other unit of the network and receives input from these same units. Recurrent networks are also called *feedback networks*. Such networks are "clamped" to some initial configuration by setting the activation values of each of the units. The network then goes through a stabilization process where the network units change their activation values and slowly evolve and converge toward a final configuration of "low energy". The final configuration of the network after stabilization constitutes the output or response of the network. This is the architecture of the Hopfield Model

2.4.2 Feed Forward Networks

These networks distinguish between three types of units: input units, hidden units, and output units. The activity of this type of network propagates forward from one layer to the next, starting from the input layer up to the output layer. Sometimes called multiplayer networks, feed forward networks are very popular because this is the inherent architecture of the Back propagation Model.

2.4.3 Competitive Networks

These networks are characterized by lateral inhibitory connections between units within a layer such that the competition process between units causes the initially most active unit to be the only unit to remain active, while all the other units in the cluster will slowly be deactivated. This is referred to as a "winner-takes-all" mechanism. Self-Organizing Maps, Adaptive Resonance Theory, and Rumelhart & Zipser's Competitive Learning Model are the best examples for these types of networks.

2.5 Feed Forward, Back-Propagation

The feedforward, back-propagation architecture was developed in the early 1970's by several independent sources (Werbor; Parker; Rumelhart, Hinton and Williams) [15]. This independent co-development was the result of a proliferation of articles and talks at various conferences which stimulated the entire industry. Currently, this synergistically developed back-propagation architecture is the most popular, effective, and easy to earn model for complex, multi-layered networks. This network is used more than all other combined. It is used in many different types of applications. This architecture has spawned a large class of network types with many different topologies and training methods. Its greatest strength is in non-linear solutions to ill-defined problems.

Rule Two: If the process being modeled is separable into multiple stages, then additional hidden layer(s) may be required. If the process is not separable into stages, then additional layers may simply enable memorization and not a true general solution.

Rule Three: The amount of training data available sets an upper bound for the number of processing elements in the hidden layers. To calculate this upper bound, use the number of input output pair examples in the training set and divide that number by the total number of input and output processing elements in the network. Then divide that result again by a scaling factor between five and ten. Larger scaling factors are used for relatively noisy data. Extremely noisy data may require a factor of twenty or even fifty, while very clean input data with an exact relationship to the output might drop the factor to around two. It is important that the hidden layers have few processing elements. Too many artificial neurons and the training set will be memorized. If that happens then no generalization of the data trends will occur, making the network useless on new data sets.

Once the above rules have been used to create a network, the process of teaching begins. This teaching process for a feedforward network normally uses some variant of the Delta Rule, which starts with the calculated difference between the actual outputs and the desired outputs. Using this error, connection weights are increased in proportion to the error times a scaling factor for global accuracy. Doing this for an individual node means that the inputs, the output, and the desired output all have to be present at the same processing element. The complex part of this learning mechanism is for the system to determine which input contributed the most to an incorrect output and how does that element get changed to correct the error. An inactive node would not contribute to the error and would have no need to change its weights.

2.5.1 Self-Organizing Map into Back-Propagation

A hybrid network uses a self-organizing map to conceptually separate the data before that data is used in the normal back-propagation manner. This map helps to visualize topologies and hierarchical structures of higher-order input spaces before they are entered into the feedforward, back-propagation network. The change to the input is similar to having an automatic functional-link input structure. This self-organizing map trains in an unsupervised manner. The rest of the network goes through its normal supervised training.

2.6 The Multilayer Perceptron Classifier

The most widely used neural classifier today is Multilayer Perceptron (MLP) network which has also been extensively analysed and for which many learning algorithms have been developed. The MLP belongs to the class of supervised neural networks.

MLP networks are general-purpose, flexible, nonlinear models consisting of a number of units organised into multiple layers. The complexity of the MLP network can be changed by varying the number of layers and the number of units in each layer. Given enough hidden units and enough data, it has been shown that MLPs can approximate virtually any function to any desired accuracy. In other words, MLPs are universal approximators. MLPs are valuable tools in problems when one has little or no knowledge about the form of the relationship between input vectors and their corresponding outputs.

The multi-layer perceptron neural network model consists of a network of processing elements or nodes arranged in layers. Typically it requires three or more layers of processing nodes: an input layer which accepts the input variables (e.g. satellite channel values, GIS data etc.) used in the classification procedure, one or more hidden layers, and an output layer with one node per class (figure 2.8).

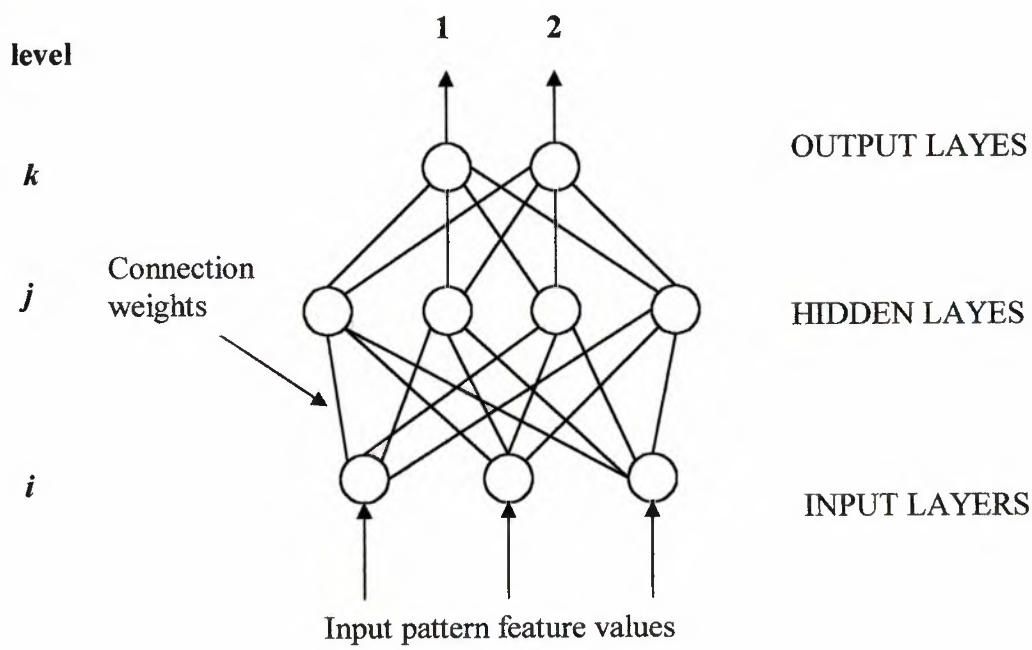

Figure 2.8 Architecture of multi-layer perceptron

The principle of the network is that when data from an input pattern is presented at the input layer the network nodes perform calculations in the successive layers until an output value is computed at each of the output nodes. This output signal should indicate which is the appropriate class for the input data i.e. we expect to have a high output value on the correct class node and a low output value on all the rest.

Every processing node in one particular layer is usually connected to every node in the layer above and below. The connections carry weights which encapsulate the behaviour of the network and are adjusted during training. The operation of the network consists of two stages. The ``forward pass" and the ``backward pass" or ``back-propagation". In the ``forward pass" an input pattern vector is presented to the network and the output of the input layer nodes is precisely the components of the input pattern. For successive layers the input to each node is then the sum of the scalar products of the incoming vector components with their respective weights.

2.7 What the Next Developments Will Be

The vendors within the industry predict that migration from tools to applications will continue. In particular, the trend is to move toward hybrid systems. These systems will encompass other types of processes, such as fuzzy logic, expert systems, and kinetic algorithms. Indeed, several manufactures are working on "fuzzy neurons."

The greatest interest is on merging fuzzy logic with neural networks. Fuzzy logic incorporates the inexactness of life into mathematics. In life most pieces of data do not exactly fit into certain categories. For instance, a person is not just short or tall. He can be kinda short, pretty tall, a little above average, or very tall. Fuzzy logic takes these real-world variations into account. In potential application of neural networks, in systems which solve real problems, this fuzziness is a large part of the problem. In automating a car, to stop is not to slam on the brakes, to speed up is not to "burn rubber." To help neural networks accomodate this fuzziness of life, some researchers are developing fuzzy neurons. These neurons do not simply give yes/no answers. They provide a more fuzzy answer.

Systems built with fuzzy neurons may be initialized to what an expert thinks are the rules and the weights for a given application. This merging of expert systems and fuzzy logic with neural networks utilizes the strength of all three disciplines to provide a better system than either can provide themselves. Expert systems have the problem that most experts don't exactly understand all of the nuances of an application and, therefore, are unable to clearly state rules which define the entire problem to someone else. But the neural network doesn't care that the rules are not exact, for neural networks can then learn, and then correct, the expert's rules. It can add nodes for concepts that the expert might not understand. It can tailor the fuzzy logic which defines states like tall, short, fast, or slow. It can tweak itself until it can meet the user identified state of being a workable tool. In short, hybrid systems are the future.

2.8 Summary

This chapter involved the understanding of the various neural networks topologies, current hardware and current software tools; the application that has been discussed here involved the selection of learning rules, transfer functions, summation functions, and how to connect the neurons within the network.

CHAPTER THREE

INDUSTRIAL APPLICATIONS OF NEURAL NETWORKS

3.1 Overview

This chapter provides a short overview of NNs in real life industrial applications. However, also within this chapter the basic features of neural networks will be shortly reviewed concentrating to the topics to be addressed in developing NN applications. Also a short review of the application areas and industrial sectors of NNs within Europe will be presented.

3.2 Neural Networks in Industrial Applications

From the engineering point of view NNs can be seen as highly parallel dynamical systems that can perform transformations by means of their state response to their input information. How the transformation is carried out depends on the NN model and its way of learning the transformation. The most natural application areas for the NNs are obviously tasks in which appropriate transformations from certain inputs to certain outputs should be established, but the transformations cannot be discovered analytically due to a variety of reasons. Therefore it is no wonder that the most successful applications of the NNs can be found in the areas of machine vision, pattern recognition, motor control, signal processing, etc., where such _inputs to outputs_ transformations dominate the problem solving.

Much of the current research in NNs is centered on individual network models, whereas in typical industrial applications, a system level view of NNs is more desirable. *Individual* NNs *are then* seen as components in a broader system which also contains many other data processing techniques, such as filtering of signals. This kind of use of NNs leads to a hybrid architecture in which some of the processing modules are based on NNs. The problem is then to decide what benefits NNs may provide for the given industrial application (if any at all) and what kinds of NN models should be used.

Up till now many different NN models have been proposed to a broad range of *applications. Also a lot of effort has been done in laying the theoretical foundations of*

NNs and the links between statistical and neural methodologies. Many NNs models, such as multilayer perceptron (MLP) or Principal component analysis (PCA) networks are similar or identical to popular statistical techniques like generalized linear models, polynomial regression, nonparametric regression and discriminant analysis, principal components and clustering. There are also some NN models, e.g. the Self-Organizing Map, that have no precise statistical equivalent, but are useful in data analysis. However, NNs should be inherently treated as statistical devices and used accordingly.

There are at least the following four main aspects that should be considered in any NN application:

1) Preparing the data: The training data should contain all the relevant information needed in building the NN model for the task. Also any a priori knowledge relevant for the problem must be considered.

2) Selecting the network model: The NN model affects crucially to the results obtained. Generalization ability, which is a measure of how well the network performs on the problem once training is complete, judges how good the network model is for the actual task. There are deferent types of NN models that can be divided to the following three broad categories according to their learning procedures: supervised, unsupervised and reinforcement models.

3) Estimating the parameters, i.e., training a network for a given problem: The task of a learning process is to construct a required transformation from the input space to the output space of the network. Any transformation of given inputs to outputs is a function approximation problem where the difficulties come from a common origin: The finite size of the training samples which easily leads to multiple possible solutions. In order to obtain useful results one must restrict the eligible solutions to a smaller set.

Coping with the constraints imposed by the data is one of the central points in the NN methodology. In addition, the effective number of parameters of the model, i.e. the NN complexity, should be matched to the problem complexity and the number of available training examples. If the network is too complex, it will perfectly learn the training set (low bias) while generalizing very poorly (high variance). Controlling the complexity is therefore a necessity to ensure good generalization. It is specially a key issue when the training set is small, noisy and even partially incorrect.

The practical methods for controlling the model complexity include, e.g., early stopped training, committees of early-stopped networks, weight decay or other regulararization methods, and Bayesian techniques for choosing the appropriate regularization level, such as the evidence framework and Markov chain Monte Carlo based methods [16].

4) Assessing the performances of the network: The general way to determine how well a network has captured the nature of a function is to validate the network with additional test set examples that were not used during the learning steps. The results obtained with the test samples can be used as indicators of the generalization ability of the network. The aim should be to determine the level of confidence in the estimated generalization abilities of the model. For this task a few useful statistical techniques can be used, e.g., bootstrap, cross validation or Bayesian treatment. Of course, the final judgement about the success of the built NN model comes in operational use.

Three application areas and industrial sectors the material of this section is based on the SIENA (Stimulation Initiative for European Neural Applications, Esprit Project 9811) project [17]. The objectives of the SIENA project were set in order to examine the current state of commercial ANN usage across the Europe.

Here we only list the most important application areas and industrial sectors of the NNs within the Europe.

Table 4.1 shows the most typical application areas of NNs across the Europe, whereas. Table 4.2 lists the most important industrial sectors. These figures are a couple of years old, but they should still quite well reflect the current situation. As can be noticed NNs have a widespread application domain across a broad spectrum of industries.

Table 3.1 Application areas of neural networks in Europe.

Control, monitoring and modeling	31%
Recognition, detection and pattern matching	14%
Forecasting and prediction	14%
Image processing	10%
Optimization	4%
Signal processing (incl. Speech and languages)	3%
Generic	23%

Table 3.2 Industry sectors of neural networks in Europe.

Production (manufacturing, agric., forest, etc.)	39%
Business services and marketing	19%
Banking, finance and insurance	12%
Medicine, health, pharmaceutics	3 %
Transportation	3 %
Utilities and energy	3 %
Wholesale and retail trade	1%
Other	20%

3.2.1 Example of industrial applications

Quality Inspection of Wood Surfaces

In this section we shortly review a quality inspection system for wood surfaces that is largely based on neural information processing principles. As a natural material wood has significant variation both within and between species, making it a difficult material for automatic grading. In principle, the inspection and quality classification of wood is straightforward: the quality class of each board depends on its defects and their distribution, as dictated by the quality standard. However, the definitions of the defects are based on their biological origin, appearance, or cause, so that the visual appearance of defects in the same class has substantial variation. The Finnish standards alone define 30 different defect classes, such as sound, dry, encased, and decayed knots, resin pockets, splits, bark, wane, mould, etc., each with various degrees of serious nesses.

(Figures 3.1 and 3.2) show examples of defects to be recognized. Figure 3.3 shows knot classes on spruce boards (from laboratory experiments during development of the system) and Figure 3.2 shows actual production line images of defects in veneer sheets. A schematic of the defect recognition system is shown in Figure 3.3. The shape related information from the defect is encoded by a Gabor-filters and self-organizing feature construction system into a “local shape histogram”. Information from the defect color is encoded as color histogram, and global features from the defect are encoded as logarithmically sampled spatial frequency spectrum over the defect. Suitable classifiers for the feature space are, e.g., subspace methods and MLP neural networks.

The approach gives knot recognition rate of about 85 %, yielding about 90 % correctness in the final grading of the boards, which is clearly better than the sustained performance of manual grading (about 70-80 %). Based on these results, an industrial machine vision based system for automatic wood surface inspection has been developed by Mecano Group Ltd, Finland [18]. The system is implemented on signal processors, with processing capacity of about 70.2 by 2 meter veneer sheets per minute. The imaging resolution is about 1x1 mm, and the sheets may contain over 20 defects on average.

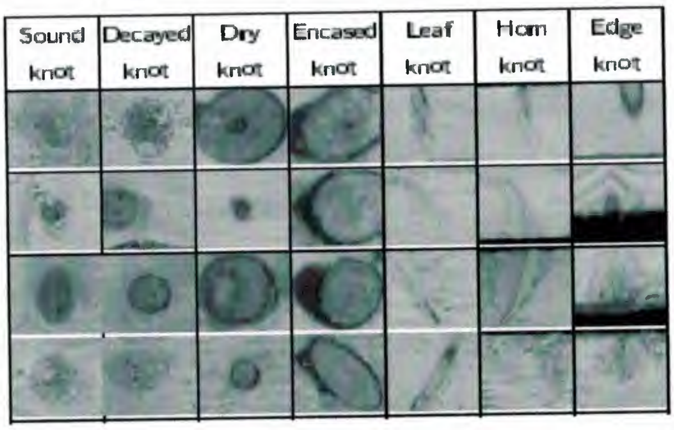

Figure 3.1: Examples of various knot types in Spruce boards (laboratory experiments.)

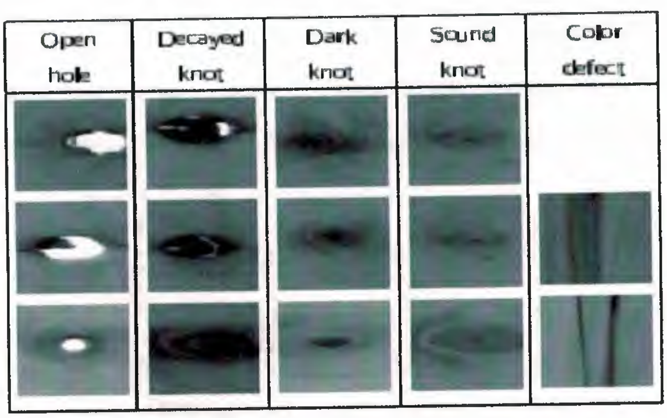

Figure 3.2 Examples of defect classes in Veneer (production line images.)

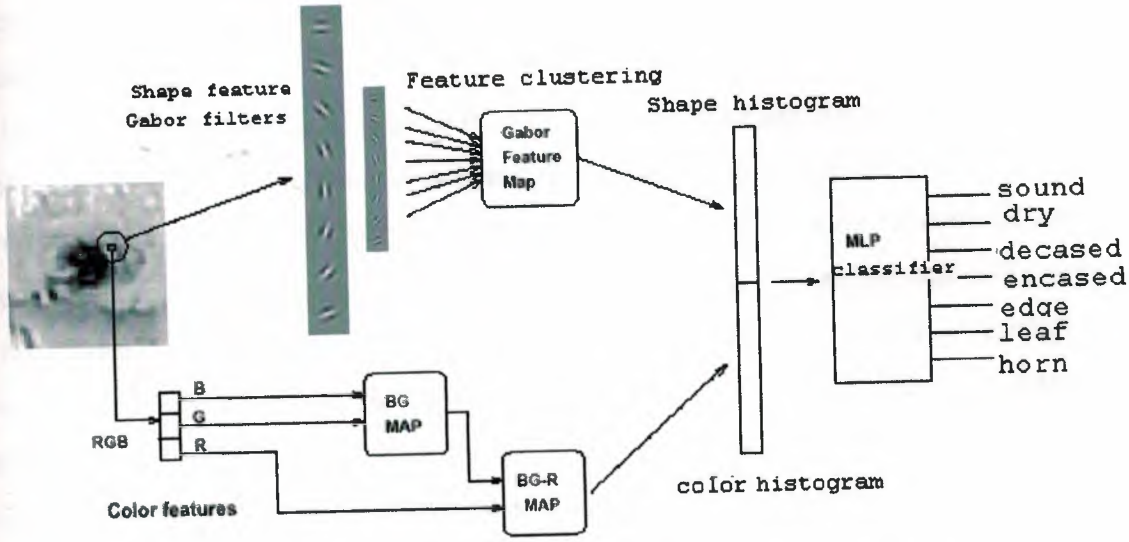

Figure 3.3 a schematic of the classification system combining Shape based and color based information.

3.3 NASA Testing New Aircraft Safety Flight Control Software

NASA is doing something to help pilots who find themselves in potentially disastrous situations flying severely damaged or malfunctioning aircraft; it is developing new "smart" software that will enable pilots to control and safely land disabled aircraft. NASA's Dryden Flight Research Center, Edwards, CA [19], is now conducting flights to test the new software which is helping NASA reach its goal to reduce commercial aircraft accident rates by a factor of five over the next 10 years.

"If an accident occurs, the aircraft will seek useable control surfaces like flaps, rudders or ailerons that would successfully compensate, restoring control to the pilot," said Dr. Charles Jorgensen of NASA's Ames Research Center, Moffett Field, CA [20], and principal investigator for the software program.

The Intelligent Flight Control System employs experimental "neural network" software developed by computer scientists at NASA Ames and the Boeing Company's Phantom Works division. When fully developed, the software will add a significant margin of safety for future military and commercial aircraft that incorporate the system. Neural network software is distinguished by its ability to "learn" by observing patterns in the data it receives and processes and then performing different tasks in response to new patterns said Jorgensen. Simple neural network software has been in use since the 1960s with computer modems to enable them to receive error-free data over often-noisy telephone lines, but it has never before been demonstrated in such a complex safety-related environment.

Using a highly-modified F-15 aircraft, the Dryden tests are demonstrating how a preliminary version of the neural network software that is pre-trained to the F-15's aerodynamic database, operating with an adaptive controller, can correctly identify aircraft stability and control characteristics, and immediately adjust the control system to maintain the best possible flight performance. The tests involve about a dozen flights over a three-week period. In its flight control application, the neural network software program takes data from the aircraft's air data sensors-airspeed, direction, pressure, force-and compares the pattern of how the aircraft is actually flying with the pattern of how it should fly. These patterns are based on a series of pre-programmed aeronautical equations or control laws that define how the airplane flies. If there is a mismatch due to equipment failures, combat damage or other reasons, the aircraft's flight control computer uses the new neural network programming to "relearn" to fly the plane with a new pattern six times every second.

For example, a military aircraft may sustain combat damage that disables one or more of its control surfaces, such as an aileron or flap. A commercial aircraft could sustain a major equipment or systems failure, such as the inability of using its flaps or encountering extreme icing, both of which could affect the safe performance of the aircraft. Using its on-line learning capability, the neural net software would identify that something has changed, and then reconfigure the flight control computer to adapt to those changes, making the failure or damage almost "transparent" to the pilot. To enable the pilot to maintain or regain control, it may change the way the remaining functional control surfaces and systems are used to compensate for the loss of the inoperative or damaged surfaces or equipment.

Future versions of the software could be developed for use in new airplanes that have digital fly-by-wire flight control systems. The system also has application to NASA's proposed Mars aircraft concept. These software versions will have even faster self-learning capability. Jorgensen noted that neural net software being developed in this NASA project could have a bearing on other aspects of contemporary life. "Once we prove neural net software can rapidly learn to fly a crippled aircraft and help pilots land it safely, then engineers will be more likely to use the intelligent software in power plants, automobiles and other less-complicated systems to avoid disasters after equipment failures," he said.

3.4 The application of neural networks to the paper-making industry

This section describes the application of neural network techniques to the paper-making industry, particularly for the prediction of paper "curl". By representing the task first as a classification and then as a regression problem, and also by calculating confidence intervals, we have made the tool (a neural network) fit the practical needs of the end-user. We present parameters characterising the current section reel as inputs to a neural network and train the network to predict whether the resulting level of curl will be within a required specification (i.e. "in-specification") a classification task. In parallel, we present these same data to another network and train it to predict the absolute level of curl, i.e. a regression task. Perhaps most importantly, we also put these two predictions in context by including confidence measures at every stage thus providing the machine operator with a powerful and insightful tool. The machine operator is then presented with a "red-light/green-light" indication of paper acceptability, a neural regression model on which the parameters can be altered to reduce curl if necessary and a clear indicator of the reliability of both diagnostics.

In limited quantities, bad curl is a significant problem, wasting plant time, engineering time and energy. This application describes the development of neural network models to encapsulate the non-linear processes underlying paper curl. These models can be used as a powerful tool for the reduction of paper curl, thus enhancing quality and reducing waste. In what has been described, in terms of processing and modelling, the approach that we take can be applied to any task involving neural networks. Furthermore, they have been developed in the context of an imperfect data collection process, typical of that found in many manufacturing operations and the combination of techniques used has particular relevance to such an environment.

3.4.1 The Database

The database provided by Tullis Russell [21] for the purpose of the work described here has a number of limitations, including missing records and measurement errors. Not least in this respect is the measurement of curl itself. Although curl is a simple quality measure, measuring curl is far from trivial. While it would seem naturally advantageous to measure curl continuously, to date this has proven to be impossible as standard techniques require the paper to be dried under controlled conditions before measurement. At Tullis Russell curl is measured after individual reels have been manufactured, leading to "out-of-specification" reels being scrapped, and a retrospective adjustment made to machine settings, according to unwritten heuristic rules developed by the skilled operators. The measurement is made using a sample of paper taken from the end of a reel and by cutting a cross-shape using a template. A glancing angle light source is then used to cast a shadow due to the curling paper at the centre of the cross. After a period of a few minutes has lapsed to allow the paper to relax the shadow is measured by hand, quantised to 5mm intervals. Therefore there may be error in the measurement due to quantisation, operator error, paper misplacement, failure to allow for sufficient relaxation time, etc... Variability in the accuracy of curl and other variable measurement could lead to significant model error and while we are developing an improved curl-measurement system, for the study reported in this section, the limitations of the database are taken as an additional constraint to the modelling process.

Various parameters are measured during the manufacture of a reel of paper. These parameters were used to classify whether the current process settings and paper specification would lead to curl that was within a required specification and additionally the level of curl that would result.

3.4.2 Preprocessing and Training

To preprocess data supplied directly from the paper-making plant a number of operations were performed. Firstly the real and symbolic data fields within the database were combined into a form that could be used for neural network training. In the case of symbolic data it is important that each field, for example the grade of the paper one-of-three for the purpose of this task, is encoded to avoid creating an artificial "weighting" to any case. Commonly a 1-of-N code is used. However, this scheme is inefficient and we use a more concise one, where the 1-of-N coding becomes 1-of-N - 1. Transforming the 1-of-N code geometrically to be the vertices of a hypertetrahedron, the codes are calculated. In this case N=3 and the transformed codes are (0.0,0.0), (1.0,0.0)and(0.5, cos(Pi/6)). This reduction in dimensionality is important in that it reduces the collinearity in the input vector, especially when there are significant numbers of symbolic parameters, which will greatly simplify the requirements of the training algorithm.

The second stage of preprocessing involved selection of the principal components within the data using the Karhunen-Lo´eve transformation [22]. For the case of the classification task the largest nine eigenvalues (or principal components) were used, while for the regression task the largest eight were used. As the principal component transformation technique is only concerned with manipulating input data, it is unable to give any insight into how important a component will prove to be in a prediction task. This information can only be determined by extensive experimentation. Therefore for the classification and regression tasks, the optimal combination of principal components was chosen through experimentation in each case. However, the transform removes any correlation between parameters and scales data to have unit variance in all dimensions, thus greatly simplifying neural network training.

To develop models for the two tasks we use Multi-layer perceptron neural networks with a sigmoidal output stage for the "in-specification" prediction and a linear output stage for the curl prediction task. The classification network contained ten hidden units and the regression network twelve. These architectures were chosen via extensive experimentation. Network optimisation was performed using Bayesian inference [23], with a Gaussian prior for the weights. We also used Mackay's evidence based framework for dynamic calculation of data noise variance and the hyperparameters defining the weight prior. This optimisation scheme also naturally allows the calculation of confidence measures. Underlying the Bayesian formalism we used a steepest descent optimisation scheme with incorporated line-search. For the regression task a least squares cost function was used, while for the classification task we used cross entropy. For both the regression and the classification task multiple networks were used and combined into a committee. The committee output was simply the averaged output from the members. For the purpose of the experiments described here, this form of committee proved to be the most reliable. In general, more complex methods of forming the committee weightings would perhaps prove successful. In the Bayesian framework the assumption is that the network weights are normally distributed and give rise to normally distributed outputs, where each output y_n is effectively the mean of the distributionand σ_n the associated standard deviation. In the committee therefore we assume that each network makes an approximation to the "true" distribution of outputs which has mean yCOM and variance,

σ^2com = $< \sigma^2_n > - < y_n >^2 + < y^2_n >$.

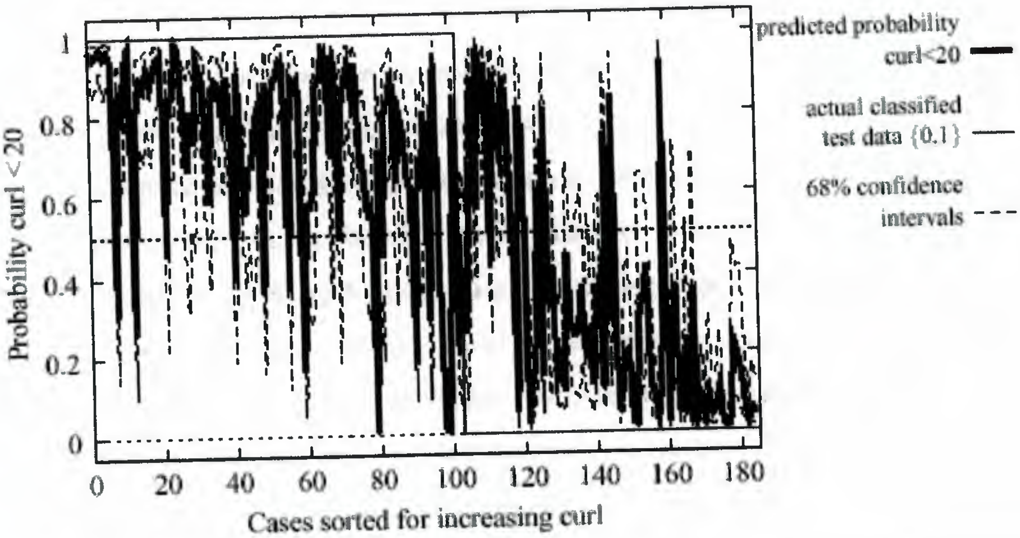

Figure 3.4: Graph showing variation in the predicted probability of curl < 20 for all cases in the test data set. The actual test data are sorted for increasing levels of curl and classified f0,1g, where "1" indicates "in-specification" (i.e. curl < 20).

3.4.3 Results

For the classifier experiments 40 networks were trained to classify the current paper characteristics as leading to paper "in-specification" or "out-of-specification". For the purpose of this experiment a level of curl less than 20 was used as the limit of acceptable curl. This level was chosen as typical. Different grades of paper have different acceptable levels. The results of these experiments are shown graphically in Figure 3.4, depicting a classification error rate of 18.92%. The test cases are sorted for increasing levels of curl and Figure 3.4 shows that the majority of the errors (where the classification boundary is set at a probability of 0.5) occur at the centre of the graph, i.e. for cases where the measured curl is 20 or near 20. The graph also shows 68% confidence intervals. Clearly the networks have been able to classify the cases to a usable degree of accuracy. Figure 3.5 shows the results for 40 networks trained to predict the absolute level of curl and tested on the same data set as above. Clearly the model has encapsulated the trend underlying these measured data, although with some imprecision. The prediction of extremely high levels of measured curl is poor. This is perhaps due to this part of the model being under-represented in the database (note that the data are densest for low curl), or that the process is different for levels of curl greater than 40. In practice, however, some accuracy in the critical 10 < curl < 30 region is most important and the predictor achieves adequate accuracy in that critical regime. The prediction of curl is therefore possible to limited but usable accuracy using the database provided and a neural network model. In addition to the curl prediction, Figure 3.5 also shows 68% confidence intervals for those predictions.

The experiments described above have tested the networks on true test data drawn from the same source as the training and validation data, but not used at all during training. The results allow us to judge model accuracy when tested with such data. In practice, however, the operator will be likely to want to see what effect changing a certain variable will have on the resultant curl. It is in this area that the confidence measures become important as it is vital to have a measure of the validity of the model. Also, within the bounds of the training data set we can expect the model to be able to interpolate between data points given a high enough data density. However, outside the bounds of the data set, extrapolating the model is more risky and this should be reflected in the confidence measures. To assess the models and confidence measures, experiments were carried out by varying a single parameter while holding the others constant and noting the prediction and the associated confidence.

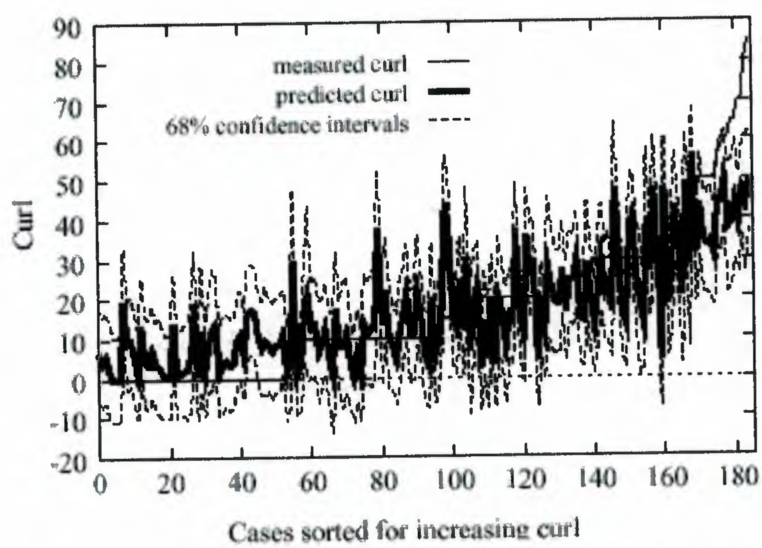

Figure 3.5 Graph showing the variation in the absolute value of curl as predicted by the committee of networks, as measured at Tullis Russell and contained in the test data.

This was done for the disparity between surface moisture on different sides of the paper as it passes through the coating machine. In addition here, rather than give the operator confidence intervals to decipher, we use our knowledge of the task and define upper and lower limit of acceptable variation, to calculate confidence as a per-centage. For the regression and classification networks the limits were set to ±10 and ±0.2 respectively. The percentage confidence may thus be calculated by integrating the normally distributed outputs, defined by yCOM and σCOM, between these limits. The results of the experiment are shown in Figure 3.6 for the classifier network. In addition to the prediction and confidence, the graph also shows data in the training set in that dimension. The input vector used in this experiment was taken from the test data set, where the original surface moisture value was 11.1, using the same scale as the graph. The graph clearly shows that as the parameter is adjusted there is a change in the output prediction and in addition that as the parameters exceed (whether positively or negatively) the bounds of the training data, the confidence falls. Clearly as this is a high dimensional problemit is unclear as to exactly how the whole data set relates to the single dimension shown, but the falling confidence is encouraging. In addition a physical interpretation of the results suggests that as the difference in the surface moisture increases in either direction so does the curl. While this interpretation also seems qualitatively reasonable we can tell nothing of the predictive accuracy of the results.

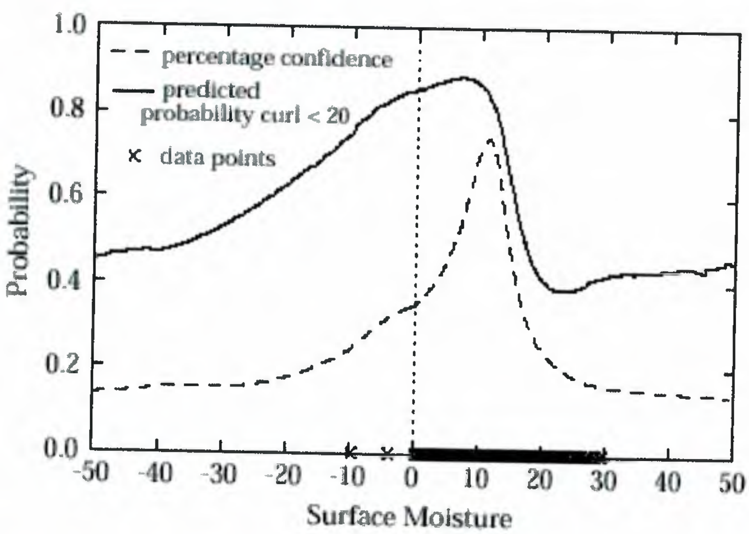

Figure 3.6: Graph showing variation in the probability of curl < 20 as the surface moisture varies. Also shown are data points in that dimension and percentage confidence.

The variance used to calculate the confidence is due to two components, uncertainty in the data and uncertainty in the model parameters. Estimating these components using evidence maximisation, as we do here only gives valid estimates when the model is used within the bounds of the training data. In terms of further work we aim to improve and generalize our estimates of these components by, in particular, including a measure of data novelty.

3.5 Summary

This chapter described some real life applications, where Artificial Neural Networks have been applied in industry. As it has been noticed that artificial neural networks has its mark in the industry.

CHAPTER FOUR

INDUSTRIAL USE OF SAFETY-RELATED ARTIFICIAL NEURAL NETWORKS

4.1 Overview

Within this chapter, it will be focused into one topic of Artificial Neural Networks in industry which is safety-related applications. Neural network products are actively being marketed and some are routinely used in safety-related areas, including cancer screening and fire detection in office blocks. As we will see how can A.N.Ns. involved in such a branch of applications such as Transport Industries and Building Services.

4.2 The Design Lifecycle

The lifecycle for computer systems with neural network components may be expressed in any of the standard ways. An appropriate representation based on that used by the FDA in their software guidance for reviewers and industry, is shown in figure 4.1

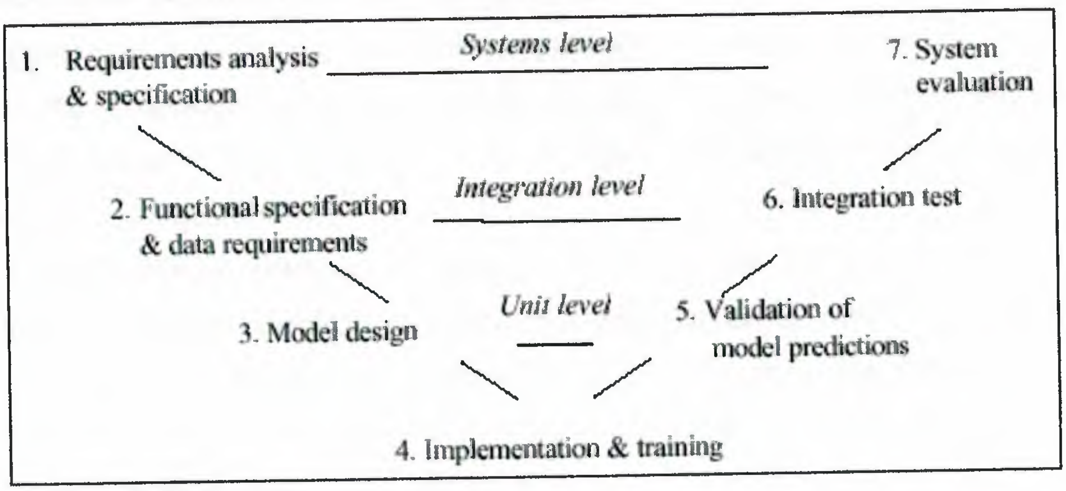

Figure 4.1 A generic software lifecycle model, with verification lines running horizontally. A neural network is an empirical model, so the model design and data requirements take-over the rôle of knowledge representation and acquisition in symbolic reasoning.

A useful approach to good practice to promote confidence in advanced computer systems is 'to integrate assurance methods that lead to high confidence into standard systems development'. This requires assurance methods to be applied at each level in the system development lifecycle. Taking as an example the design of a non-linear process control module, at the unit level, controllers must be validated against established principles of stability margins and performance bounds. Extending the assurance principle to static pattern recognition, a performance assessment of the neural network implementation requires confidence estimates for the network predictions, possibly supplemented by novelty detection, which affords more specific protection against extrapolation from the design data. These requirements loosely reflect the concerns with completeness of the knowledge base and consistency of inference, familiar from prepositional logic.

It follows that some of the critical issues of particular relevance to neural networks are the need to focus in areas where significant data are available, difficulties with scaling systems from prototypes to routine use, and evaluation on real data. There is evidence that data collection, pattern representation and data integrity checking can take up a large proportion of the development time. A joint report by the Neural Computing Research Group at Aston University and Lloyd's Register identified [24], among substantial issues for assessment of neural networks, the need to characterise data quality, relate this to defined safety margins, and eventually to trace neural network design features to the overall specification requirements at the system level.

At the integration and systems levels, the hazard analysis converges with the standard approaches. However, there are specific issues for systems with neural network components. To quote NASA [25]:

'Current software practices are human-centred activities. Such practices do not address very large or complex systems [to the extent that] novel software architectures using artificial intelligence and neural networks are stretching the behaviour understanding that can be achieved through state-of-the-art software validation and verification technologies.'

These points to the need to re-evaluate, for computational intelligence, how to impose discipline on design to minimise the introduction of additional complexity. With neural networks, at the unit level this requires parsimonious designs, and the integration level it involves reconciling the empirically derived optimal responses, with the structural models more naturally suited to represent expert knowledge. Given that 50% of errors are introduced at the requirements stage, a related issue is how to take a system-level view including human factors. In the same report, the FDA calls for a set of expected best practices and complete hazards analysis of information-based safety-related systems.

The lifecycle is much the same as for decision systems involving statistical modules although for neural networks the technical aspects of verification at the systems, integration and unit level, are not yet established. There are also parallels with the corresponding stages in the design of knowledge-based systems, and indeed to any inference system with substantial non-linear components, whether using symbolic or distributed knowledge representations.

Tracing through the blocks in figure 4.1:

1. Expressing system requirements involves specifying against unwanted behaviour, in responses to unforeseen sequences of events. Many applications are now targeting environments that cannot be regarded as closed, and for which knowledge representations will necessarily be incomplete. Medical diagnosis is an early example of this.

2. Knowledge representations impact on generalisation ability, i.e. correct operation for future situations. In particular, human expertise is not always consistent and complete, and can be difficult to capture into an algorithmic representation.

3. It is interesting that there appears to be a convergence of knowledge-based, neural computing and statistical modelling approaches. The focus is on Bayesian models, where prior knowledge is used to define structural relationships, and statistical models are situated at the nodes.

4. Assessing convergence is the network equivalent of achieving consistency. It can be established on the basis of statistical or computational learning criteria, to ensure appropriate coverage of the data without over-training.

5. There has recently been some controversy over the development of statistical models for medical diagnostics, emphasising the need for independent assessment by agents external to the original design process. This is partly the need to ascertain and automatically signal if the inference is extrapolating outside, rather than interpolating within, the knowledge base. This is a well know issue for neural networks, and while error bars go some way towards addressing novelty detection, it remains an acute problem when parsimony due to model selection causes data from none of the classes to map into highly predictive regions from one of the assumed classes.

6. Transparency of inferences is difficult for any complex system, and particularly so when knowledge is distributed. Nevertheless, in most current applications the optimised neural networks are sparse in the number of input variables used. Their small size allows their operation to be sufficiently traceable by direct inspection of the weights and hidden node activations in response to specific test patterns, to enable a verification against established domain expertise.

7. The crux of software regulation may be summarised as validation, verification and testing (VV&T). Any system may in principle be consistent and complete by design, yet contain knowledge that is incorrect. Therefore, while VV&T requires adherence to formal methodologies at each level of the design lifecycle, where non-linear inferences from real-world data are involved the emphasis appears to be shifting towards extensive trials with external data. This leads to a verification process that is performance-based, rather than founded on internal logic. A key element here is the need to define the required performance targets at the specification stage, rather than continually adjusting them on the basis of results obtained with network prototypes.

Progress in neural networks relating to the issues raised in each step of the lifecycle is discussed later in the report. In relating neural networks to other artificial intelligence methods, it is useful to place them also in the wider context of non-linear signal analysis, whose rôle they are sometimes designed to implement. The schematic in figure 4.2 extends the discussion to a wider spectrum of models for information processing, on the basis that all of the methods listed involve selecting information for a parsimonious inference model. In this selection process, much information is lost, and the theoretical principles behind the system design and analysis are there to provide assurances about the accuracy and consistency of the predictions made in response to future queries, whether expected or novel, falling within or outside the system specification.

In most practical applications in the process industries, a continuum of models is used as a cascade, from conventional to novel engineering analysis, through core neural network systems, to supervisory knowledge-bases systems enforcing some form of self-validation.

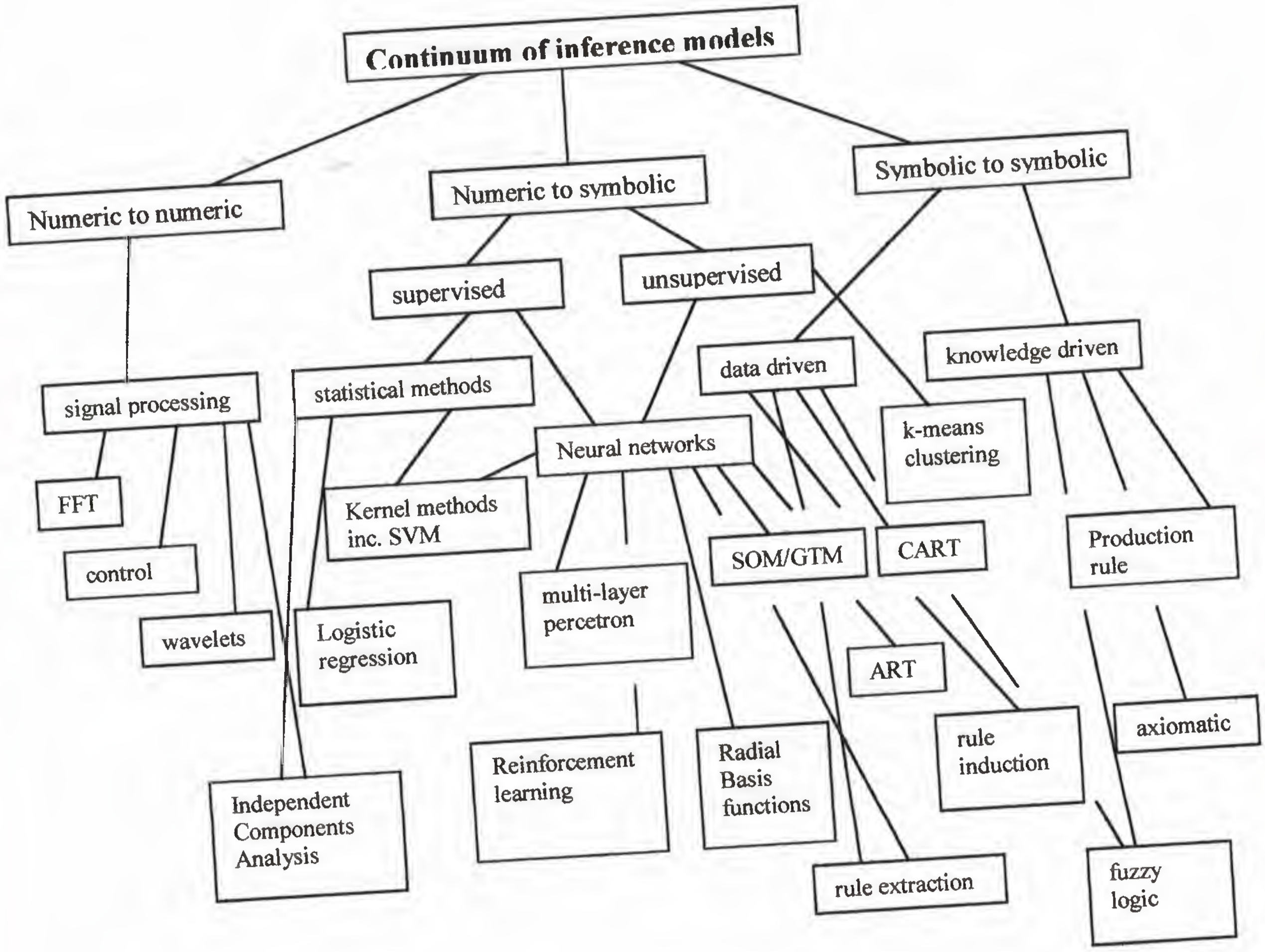

Figure 4.2 Global context of inference model. Some of the assignments are necessarily ambiguous and the list is not exhaustive, e.g. Bayesian models with latent variables straddle all three main headers.

4.3 Power Generation and Transmission

GNOCIS (Generic NOx Control Intelligent System) was developed by Power Technology for use as an on-line advisory or closed-loop supervisory system for NOx emissions [26]. The purpose of the software is to adapt to long-term changes in the plant condition, enabling better optimisation of the operating mode of the plant. Trials were conducted at a 500 MWe unit at Kingsnorth power station, claiming to identify major annual efficiency savings by reducing carbon-in-ash, worth more than £100k per annum, while maintaining NOx emissions under prescribed limits. GNOCIS has now been applied to a range of boiler sizes, with several close-loop applications, with substantial efficiency gains.

In a separate project jointly funded by BCURA (British Coal Utilisation Research Association) [27] and the Department of Trade and Industry, a hybrid Neural Network based controller for a 3.7 MWth (i.e. MW thermal) chain-gate stoker-fired shell boiler at Her Majesty's Prison Garth, Leyland, in collaboration with James Proctor Ltd. This demonstrated 10% lower NOx emissions without sacrificing carbon-in-ash losses and with a 10% reduction in CO emissions, in steady state, with gains also when load-following. The code was implemented in Matlab, and expert knowledge had a key role for the integration of the Neural Network module into an efficient control loop structure.

A related field with potential in power generation is soft sensing, meaning software to estimate specific process variables that cannot be directly measured, for instance pulverised fuel flow to the boiler. In control terms, soft sensing represents a non-linear extension of linear observers, which are typically Kalman filters. In addition to efficiency gains from tighter control this measurement would also provide fast indications of dangerous conditions, such as mill emptying by coal sticking to the feeder chains. Neural network based optimisation is recognised as a generally applicable technique with potential benefits for several continuously monitored loops in power stations.

Load demand forecasting is an early focus to demonstrate the potential of Neural Networks in practical problems from the power industry. There are at least two commercialised load demand forecasting systems, namely Nostradamus, from NewEnergy Associates and ANNSTLF (Short-Term Load Forecaster). These software packages are used for daily and hourly electricity and gas demand forecasting. In the case of National Fuel Gas Distribution Corporation, in Buffalo, the forecast is for mode than 730,000 customers in Western New York and Northwest Pennsylvania. ANNSTLF claims to be implemented at over 30 electric utilities and to be used in real production environments, promising to cut operating and fuel costs by 5%. Since 1995 the Pennsylvania power & Light company has used ANNSTLF as its only forecasting tool for power sales. Reductions in the "power cushion" is projected to save the utility US$4 million in the next ten years.

Accuracy and economic value of Neural Networks for short-term electric demand forecasting, and for combustion optimisation with reduced NOx emissions, both featured at the 1998 America Power Conference [28], and the 1999 International Business Forecasting Conference. Savings achieved by Duke Power Company in its inspection of reactor core control assemblies (RCCAs), from time reductions in the analysis of the 800 Mbytes of data resulting from a single core inspection, are estimated at $28,000 per inspection. This is because only 5% of the data contain relevant wear information. Together with the Knowledge Based Technology Applications Centre (KBTAC) of the Electric Power Research Institute (EPRI) Duke Power [29] developed an Intelligent Data Reduction and Processing system (DEK-IDRP) that uses Neural Networks and rule based expert structures. The emphasis in the DEK-IDRP is on near-real-time advice with user-friendly GUIs, and an evaluation has claimed the detection of previously missed wear features. The operation of this system to analyse data from pressurised water reactors is expected to save 'US$361k in the next 5 years'. This application is typical of a staged approach to safety-related automation, where an advisory system is used simply to flag-up potentially important information, leaving the user to decide what information is actually important. This philosophy of operation is similar to that employed by Papnet, both systems having the rule of helping human experts to sieve through a mountain of information, looking for a small number of tell-tale signs. This high sensitivity comes at a price of low specificity, which is resolved by the involvement of the user.

Nuclear reactor surveillance and diagnosis was the subject of a neural network benchmark study carried out in 1995 under the auspices of the OECD-NEA. In a separate project funded by the US Department of Energy, an artificial intelligence fault-diagnostic system for real-time detection of component failures was developed with data from Duane Arnold Nuclear Power Station. The OECD-NEA benchmark was to predict the generated electrical power from a two-loop pressurised loop reactor at Borssele in the Netherlands. Some accuracy was reported except during condenser rinsing and stretch-out during shutdown operation. However, there was a recognition that a MLP trained with the data collected from plant does not adequately capture the process dynamics, especially when it targets one-step-ahead prediction of process output from the control variables. This is now well accepted, leading several groups towards using non-linear combinations of standard linear dynamic models, which can be identifie d locally with analogues of PRBS. In a follow-up report a year later, it was concluded that there are 'serious limitations to the effective utilisation of the MLP in practice', and that the benchmark results 'reveal the importance of establishing general guidelines for enhanced network training and effective utilisation'.

4.3.1 Process Industries

A manufacturing area where neural network control has been successfully applied for some time is steel rolling mills. Having developed prototype neural-network models for strip temperature and rolling force at the hot strip mill of Hoesch, in Dortmund [30], in 1993, Siemens has applied this technology at 40 rolling mills world-wide. Claimed efficiency gains are 30% better accuracy in rolling force modelling, and with prediction improvements leading to US$200k p.a. in material costs. The considerable business benefits demonstrated in routine use by a major manufacturing company, are indicative of the industrial push to explore new, non-linear, technologies. This application also illustrates two generally applicable exploitation areas for Neural Networks in process control. The first is to interpolate parameter settings more accurately than is possible with inheritance tables. This serves to close the gap between principled analytical models, and the effect of unmeasurable parameter disturbances, the network providing correction factors, in effect to calibrate the analytical model from batch to batch.

The second growth area to automate systems of increasing complexity is perhaps comprising a number of interacting units, and often involving switching between different operation regimes. It is interesting to relate Siemens' experience that standard MLP-type feed-forward networks were not successful and purpose-built models for dynamic control had to be designed. These models are claimed to allow robust on-line adaptation, compensating for process drift even in multivariate processes.

Returning to the development lifecycle, Siemens indicates that Neural Networks always complement, and never replace, physical models. This statement implies that returning to basic principles is a key to the interpretation of the non-linear models generated from the data, and hence critical to the safe, as well as efficient, process operation.

Secondly, domain expertise, here in the form of linear and non-linear process control theory, is essential in the validation process.

Thirdly, the data requirements are severe, taking thousand of strips, over several weeks' production time, to build a representative data set. All of these factors point to the issues that may be involved in verification of Neural Networks in safety related process control applications.

At Siemens, 'intelligent controllers' have also been applied successfully to coat thickness in hot dip galvanising lines. Furthermore, there are indications that these methods may be applied to the design of traditional proportional integral and derivative (PID) controllers for plant with time delays, which would considerably broaden the scope for their application.

A generic third strand of application is in fault detection and identification (FDI). While frequently not a safety-related area, nevertheless it, too, has the potential for widespread use of non-linear dynamic models, as well as static and self-organising Neural Networks, serving various predictive functions from which departures from normal operation may be inferred. Despite some theoretical work in this area, visualisation of process variables remains fraught with difficulties because of the often severe dimensionality reduction that is required, which is exacerbated by the expedient dependence on heuristic methods. In contrast with soft sensing, which targets a specific process variable that is hidden from direct measurement; fault diagnosis often demands the simultaneous monitoring of a large number of variables, to detect departures from an ill-defined normality envelope. It is, in essence a novelty detection problem.

4.3.2 Transport Industries

Aircraft icing is a major hazard for which weather forecasters must advise pilots. The Experimental Forecast Facility at the Aviation Weather Centre in Kansas City, Missouri, is currently evaluating NNICE, a neural network-based icing intensity predictive forecast tool [31]. Also in the US, at NASA's Dryden Flight Research Centre, Edwards, a joint programme with Boeing is testing neural network damage recovery control systems for military and commercial aircraft [32]. The purpose of the research is to add a 'significant margin of safety' to fly-by-wire control, when the aircraft sustains major equipment or systems failure, ranging from the inability to use flaps to encountering extreme icing. Example aircraft where this approach can be applied are the Boeing 777, and the current test plane, a F-15 with canards and pitch/yaw vectoring nozzles.

At Long Beach airport inductive loops are used to identify aeroplanes at specific locations on the runways, using Loop Technology (LOT). The potential for use of low-cost surface sensors in avoiding incursion incidents relies on Neural Networks to classify loop induction signatures for accurate aircraft type identification. In the UK, vibration analysis monitoring in jet engines is the focus of a research project involving Rolls-Royce and the Department of Engineering at Oxford University. This produced a diagnostic system, Quince which combines the outputs from Neural Networks with template matching as well as statistical and signal processing methods, processing them with a small set of rules.

The software is designed for the pass-off tests of jet engines, has a tracking facility to suggest the most likely fault, and centres on the use of novelty detection to identify unusual vibration.signatures. According to the web site, Quince is now being licensed to Rolls-Royce under the terms of a Licensing Agreement signed in May 1998.

A second condition monitoring application between Rolls-Royce and Oxford University involves predicting a thermocouple reading of the exhaust gas temperature in aero-derivative gas turbines with a power output of 3-50 MW. High prediction errors are indicative of developing faults, and it is claimed on the web site that the model is capable of identifying real faults several hours before it is detected by the control-system logic which shuts-down the engine.

A third collaborative application is to perform comprehensive whole engine data analysis and interpretation, with attached confidence levels, by fusing diverse sensor readings (performance parameters, vibration spectra and oil debris information) to produce 'reliable indications of departures from normality'. The aim is real-time in-flight monitoring for the new Trent 900 Rolls-Royce engine. Technically the project combines standard observers, i.e. Kalman filters, with more advanced signal processing techniques and Neural Networks, as well as other elements of computational intelligence.

European collaborative projects in Framework Programmes four and five have implemented Neural Network control demonstrators, ranging from engine management models to physical speed control, and involve leading car manufacturers. The key feature of these control systems is the combination of detailed engineering expertise with non-linear interpolation by neural network architectures. In specialist applications such as real-time control of complex dynamical systems, sometimes requiring rapid adaptation to operational environments, and always susceptible to re-tuning for production models and over time during routine maintenance, it is likely that dynamic models designed from first-principles will be at the core of controller design. An important consequence of this approach is that closed-loop stability can be theoretically established, even when switching between different operating regimes, themselves operating under the guidance of supervisory neural-, fuzzy- or rule -based systems. This is a good illustration of the smooth coupling of a range of inferencing methodologies, that is key to many important practical applications where novel solutions are needed. In process control generally, stability can now, in principle, be assured for complex combinations of very different models, operating in the real-world with a wide range of disturbances and uncertainties.

4.3.3 Building Services

Siemens currently markets the FP-11 intelligent fire detector. This uses 'FirePrint' technology, which is based on fingerprints, that is to say time traces from different types of sensors. These were acquired from fire tests carried out over many years, resulting in extremely high specificity, triggering one-thirtieth as many false alarms as conventional detectors. The developing company, Cerberus, a division of Siemens Building Technologies, are sufficiently confident about this product, also called AlgoRex, that they have offered a refund for the costs of any unnecessary fire department visits triggered by this alarm. The detector's appearance is similar to an of-the-shelf fire detector, mounted onto a bulky base. It offers three user-selected options for the system logic, using an optical sensor to detect smoke density, a heat sensor for temperature, or the neural network which combines both measurements. In reality, the network is based on a digital implementation of fuzzy logic, with rules discovered by the neural network but validated by human experts. According to the manufacturers, 'the rules used in the fuzzy logic system are the result of decades of know-how gathered by experts at Cerberus from approximately three million AlgoRex fire detection systems in operation world-wide'..A separate product under development is the WaveRex flame detector, which operates in high ceiling sites where the amount of smoke would not necessarily trigger a detector. This combines a fuzzy logic system with a wavelet filter bank, to distinguish profiles of real fires from those of reflected sunlight. Flicker measurements are made at 4.3 mm (CO2), 5 and 6 mm (black-body radiation) and 0.8 mm (visible range). The product sheet claims that this is the first time that wavelet analysis is used in a mass produced article.

A further generic area where there is much demand for new products is smart buildings, which extends from semi-automation of household appliances to remote care for convalescing or elderly patients. At Siemens, active vision is being developed, through cameras with built-in computers that capture event profiles for automatic detection of the event even under variable lighting conditions. Neural networks are likely to be at the centre of this.

4.4 Potential Safety Gains from Neural Networks

Although applications of Neural Networks range across a large number of industries, in the vast majority of current applications their rule fits into two generic classes, each with further specialisations.

4.4.1 Static Pattern Recognition

This is the original motivation behind the exploitation of neuromorphic systems, but developments have forked into two distinct routes. The vast majority of image processing, sensor fusion and diagnostic systems use static Neural Networks whose function is entirely statistical by nature, and whose rule is that of distributed, non-linear, associative memory models. When they are regarded in this manner, their benefits, associated hazards, and regulatory requirements could be expected to align with those for linear multivariate statistical methods, but the theory underpinning robustness is much less developed for non-linear than for linear statistics. Potential safety gains are to be derived form automation of diagnostic and monitoring systems, as well as greater accuracy, usually expressed in terms of very specific performance requirements for true detection rate (sensitivity) and false alarm rate (specificity).

A second strand of current developments follow-on from our nascent understanding of neuromorphic systems, which is the case of mammalian sensory networks have become very detailed. For instance, the physiology of the retina is sufficiently well understood at the level of axonal signals, to enable electrical replicas to be constructed, whose dynamic range and stability under variable light conditions are extremely good. Beyond this, at a cortical level, simple and complex cortical cells have been directly mapped for their frequency response characteristics, and can be emulated using banks of wavelet filters combined into quadrature pairs. A commercial example of this is an iris classification system for access control. With wavelet-based whole face recognition it is possible even to characterise facial expression. Auditory and olfactory systems have also been modelled in detail, and their performance potential is huge. Potential safety benefits arise from the ability to place near-human performance in locations or environments where it is either impossible, or too costly, to employ people. As the hardware implementation of 'bionics' develops further, there is clear potential for prosthetic implants, as well as the design of circuits that exceed certain aspects of human sensory performance, notably speed of response. An example is the development of artificial retinal implants.

Further practical benefits from emulators of cortical circuits are sparse signal representation and excellent signal-to-noise differentiation. It is not difficult to conceive of safety-related applications reliant on 'bio-sensors' coupled with 'intelligent' supervisory systems, for instance for fire or smoke detection in remote environments, in the presence of variable conditions such as exposure to inclement weather, requiring neuromorphic processing to maximise the contrast between the signal sought and a background of continually varying disturbances. Yet another potential gain is in human-computer interaction, where hand-written and verbal natural language interfaces are already being developed.

4.4.2 Dynamic Control and Monitoring

Immediate performance benefits can be gained where control with complex switching requirements is optimised by replacing parameter allocation look-up tables with smoothly interpolating non-linear models. In many cases this requires identification and control of dynamical systems, a very difficult task where non-linearities are involved. Recent developments have resulted in theoretical frameworks to demonstrate stability for non-linear control, and these have arisen from combined developments in control theory and in numerical analysis. It is a telling development that traditional control expertise is even more necessary for the safe deployment of neural network controllers then it was for traditional three-term controllers. The need for this expertise is apparent from a constructive approach to the development lifecycle. At the unit level, confidence should be built-in through design with stability and performance assurances. At the integration level, the availability of advanced controllers may enable, for instance the replacement of hydraulic actuators with electrical ones that are easier to build back-up systems for. At the system level, there are implications arising from the additional flexibility afforded by, for instance, drive-by-wire, where braking and suspension control can be inter-linked to correct lateral stability under heavy braking, to give but one obvious example. In another example, taken from the preceding section, fly-by-wire control may be automatically reconfigured to compensate for severe system failure. This applies equally well to commercial, as to military aircraft. Ultimately, safety benefits will be gained from tighter control of complex switching systems which exceed human performance in emergency situations.

Further engineering gains are likely to result from developments in intelligent sensors, for which demand from industry is substantial. Safety benefits will arise from access to estimates of hidden state variables. An example of this would be estimating the amount of pulverised fuel in a coal mill, when wet coal can become trapped in the rolling chains of a coal feeder. This results in fast emptying of the mill by the primary air flow, which is there to dry the coal, filter fine particles in a cyclone, and carry the coal directly to the burners. The measurement difficulty is that any probe to measure pulverised fuel flow quickly clogs-up, yet this measurement is important both to regulate the fuel efficiently, and for safety, since an overly rich air-to-fuel ratio is spontaneously combustible at the temperatures that these mills reach when running empty. In effect what is needed is a sophisticated non-linear observer capable to estimating the coal content of the mill for air temperature and differential flow measurements. This is just one example from many in the process industries where direct measurement of variables key to performance and to safety are either too expensive to measure accurately, or even impossible to measure fast enough for closed-loop control, for instance pH values in sulphonation loop reactors. Plant monitoring is altogether a different type of system, where the aim is to compare actual plant operation with nominal operation given the current control actions. Any deviation is indicative of plant malfunction, which can trigger a diagnostic search for the most likely fault, in real-time. The benefit here is in earlier warning of incipient faults, as well as the potential for providing the operator with a ranked list of possible hazards, replacing a plethora of consequential alarm calls. The nuclear sector is a potential area for advisory systems of this nature, but the potential is much wider that this. The difference between using Neural Networks rather than purely knowledge-based systems for this application lies in the ability to integrate complex signa ls at a low level, yielding more complex advice than may be possible by representing the activity of individual sensors directly in symbolic form.

4.5 Summary

This chapter carried out on its content one of the most important issue in any field within the industry which is Safety. In here the chapter descussed several aaplications in industry to show how important and significant this is, applications such as Transport Industries and Dynamic Control and Monitoring. Artificial Neural Networks was that much effective in our life, as it has been described in here, in safety-related areas, including fire detection in office blocks. Safety benefits are claimed from improved performance, for example better specificity in alarms.

CONCLUSION

The research and application sectors of NNs are constantly working towards a more comprehensive understanding of NN models and their strengths and shortcomings in a wide range of applications. It seems that the past over exaggerated claims of the capabilities of NNs have become more sensible and realistic, i.e., the field of NN has evolved to correct direction and matured. Realistic expectations combined with positive feedback from the applications have also reduced the danger of NN interest to collapse due to over- realistic and unfulfilled promises. Nowadays it is known quite well what NNs are and what they are capable to do. When this knowledge is combined with industrial expertise needed in building a successful application, the results should be at the level of the NN capabilities and also objective.

Chapter One: in this chapter it has been focused on the various definitions of the Artificial Neural Networks, what are the Artificial Neural Networks, a brief history about A.N.Ns. moreover, what made A.N.Ns. differ from traditional computing and expert systems, as whats the goals of using A.N.Ns. in real life applications, whether there is any limits to the technology of Artificial Neural Networks and the future of it and what applications could be involved in have also been retrieved within this chapter.

Chapter Two: within this chapter the archeticture of Neural Networks and some of its topologies had been shown and also discussed; the way that A.N.Ns. is behaiving within the teaching the N.N. session has also been illustrated, till the next development.

Chapter Three: in here within this chaper specific real life applications of N.Ns. these applications were in industrial sectors, so some of these applications were: the quality inspection of wood surfarces, NASA developed a new software which is going to take the plane or the aircraft to a safe land if something fault happend which may cause damage "disaster" which threatens people's life on board.

Chapter Four: was about one topic or one industrial application, as described within chapter three N.Ns. were involved in safety within the indutrial applications, so here Industrial use of safety-related artificial neural networks with many applications which have been retrieved and illustrated as well.

While the search and preparing the topics for the project, I have come to various kind of information and knowledge, but what was so important is, how to be accurate and fussy.

The overall objectives of this project are to discuss to what extent neural networks are used, and are likely to be used in the near future applications. In detail:

- To describe and categorize or distinguish current industrial applications of neural networks.
- To discuss the issues arising from current industrial use of neural networks, including: the difficulties in achieving and demonstrating safety within industry;
- To indicate other key centres of excellence in neural networks and their application.

I have got a lot of knowledge about the Artificial Neural Networks and their applications, especially in industry therefore, so here, I can say that A.N.Ns. will come up with new solutions to difficult issues where other technologies are unable to solve it.

REFERENCES

[1] F. Rosenblatt, Principles of Neurodynamics. New York: Spartan Books, 1962.

[2] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation", in Parallel Distributed Processing.

[3] P. Werbos, Beyond regression: new tools for prediction and analysis in the behavioural sciences.PhD thesis, Harvard University, Cambridge, MA, 1974

[4] Varjo, J., ``Controlling Continuously Updated Forest Data by Satellite Remote Sensing", International Journal of Remote Sensing, 17(1), pp. 43-67, 1996

[5] According to the DARPA Neural Network Study (1988, AFCEA International Press, p. 60)

[6] Haykin, S. (1994), Neural Networks: A Comprehensive Foundation, NY: Macmillan, p. 2.

[7] A general feeling of frustration (against neural networks),1996.

[8] From the World Wide Web: www.gis.uiuc.edu/

[9] From the wolrd wide web: www.dacs.dtic.mil/techs/abstracts/neural.html

[10] Grossberg, Stephen, "A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine", *Computer Vision, Graphics and Image Processing 37*, 1987.

[11] From the World Wide Web: www.dacs.dtic.mil/techs/neural/neural_ToC.html

[12] C. M. Bishop. Neural Networks for Pattern Recognition. Oxford University Press, 1995

[13] From the World Wide Web: http://www.mbfys.kun.nl/SNN/siena

[14] J. Lampinen, S. Smolander, and M. Korhonen. Wood surface inspection system based on generic visual features. In F. Fogelman-Soulié and P. Gallinari, editors, Industrial Applications of Neural Networks, pages 35_42. World Scientific Pub Co, 1998.

[15] From the World Wide Web: www.usc.edu/dept/engineering/TTC/NASA/newsarchives/apr_99control-software.html

[16] E.T. Langevin and W. Giguere. Online curl measurement and control. TAPPI Journal, 77(8):105–110, 1994.

[17] C.M. Bishop. Neural Networks for Pattern Recognition. Oxford University Press, 1995.

[18] D.J.C. MacKay. Bayesian framework for backpropagation networks. Neural Computation.

[19] From the World Wide Web: www.ncrg.aston.ac.uk/cgi-bin/tr_avail.pl?

[20] | From the World Wide Web: www.ccic.gov/pubs/hcs-Aug97/hcs_problem_views.html.

[21] From the World Wide Web: www.powertech.co.uk/enterprise/r&d_portfolio/gnocis_art.html

[22] From the World Wide Web: www.dti.gov.uk/ent/coal

[23] From the World Wide Web: www.nuc.berkeley.edu/thyd/ne161/rtse/dukerods.html

[24] From the World Wide Web: www.awc-kc.noaa.gov/awc/nnice.html)

[25] From the World Wide Web: www.usc.edu/dept/engineering/TTC/NASA/newsarchives/apr_99control-software.html