NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

IMPLEMENTATIONS OF NEURAL NETWORKS

Graduation Project
COM- 400

Student: Mahmoud Said Ashour

Supervisor: Assoc.Prof.Dr. Adnan Khashman

Nicosia - 2002

ACKNOWLEDGEMENT

"First, I would like to thank my supervisor Assoc.Prof.Dr Adnan Khashman for his great advice and recommendations to finish this work properly. Although I faced many problem collecting data but he has guiding me the appropriate references. (DR Adnan) thanks a lot for your invaluable and continual support.

Second, I would like to thank my fiancée (Noura) for her emotional support and encouragements.

Third, I would like to thank my family for their constant encouragement and support during the preparation of this work specially my lovely brother (Asad).

Fourth, I thank all the staff of the Faculty of Computer Engineering for giving me the facilities to practice and solving any problem I was facing during working in this project.

Fifth I do not want to forget my best friends (Nurdan) and (Fatos) for helping me to finish this work in short time by their invaluable encouragement.

Finally thanks for all of my friends for their advices and support specially Majdi Ashour, Hafez Abu Jabal, Esa Anan, Mohmmed abu Ammra, Haider Al Agha, and many thanks for my lovely brother Moheeb al Qomboz for his special encouragements."

ABSTRACT

Artificial neural networks (ANNs), also called parallel distributed processing systems (PDPs), and connectionist systems, are intended for modeling the organizational principles of central nervous system, with the hope that the biologically inspired computing capabilities of the ANN will allow the cognitive, and sensory tasks to be performed, more easily and more satisfactorily than with conventional serial processors.

ANN structures, broadly classified as recurrent or non-recurrent, have numerous processing elements (also dubbed neurons, neurodes, units or cells…etc).

Neural networks are trained by repeatedly presenting examples to the network. Each example includes both input and outputs.

Neural networks have many advantages; such as memory and processing elements of the network are collected, self-organizations during learning, parallel and asynchronous, and cycle-time in millisecond.

Neural networks inspired by information processing strategies of the brain, are proving to be useful in a variety of applications, such as manufacturing, science, medicine … etc.

Neural networks are applicable for pattern recognition, such as decoding algorithms, organizing chemical drawings; many resources are needed for these applications to be used.

The project reports on new applications of neural networks in real life, thus it can be obvious that the neural networks are applicable in many fields. Neural networks take the mind to imagine what ever can't be clear and trustable of the day's technologies, such as OCR contents providing, voice mail recognizing, stocks, business and bank administration.... etc.

INTRODUCTION

There has been a proliferation of literature from scientists in several disciplines on the topic of neural networks since its resurgence in the early 1980s. 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. Many 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, relatively few researchers made important advances.

Chapter one describes the background of neural networks, and the history and principles of neural networks, how the neural networks are being improved from 1940 to be the great topic of the technology fields. Neural network is defined as a system composed of many simple processing elements operating in parallel, whose function is determined by network structure, connection strength, and processing performed at computing elements.

The idea of neural networks has been around since the 1940s but only in the late 1980 were they advanced enough to prove useful in areas such as computer, control, and speech recognition.

Neural networks are unlike artificial intelligence structure in that they trained to learn relationships in the data they have been given, just like a child learns the difference between a chair, and a table by being shown examples.

Neural networks can be trained using additional input variables, once trained they are very fast; they deal with non-linearity in the world in which we live. They handle noisy or missing data, finally they can provide general solutions with good predictive accuracy.

Chapter two describes the architecture, and basic components of neural networks. The most basic components of neural networks are modeled after the structure of the brain. Neural networks name have a strong similarity to the biological brain, and therefore a great deal of the terminology is borrowed from neuroscience.

Models of our own brains, and nerve cell motivate neural networks architectures. Although the knowledge of brain is limited, we do have much detailed anatomical, and physiological information, the basic anatomy of an individual nerve cell is known as (neuron).

Neural networks have many meanings of term “architecture” depending on whether you are talking about buildings, inside of computers or neural networks among other things. So when we refer to such a word “architecture”, it means the set of interconnections and learning a logarithm defined for it.

Neural networks can be classified according to the way they learn. Learning can be performed on a supervised or unsupervised learning. Supervised learning algorithm can be such as backpropagation, and Hopfield network. Unsupervised examples can be such Kohonen’s learning, competitive, and Adaptive Resonance Theory.

Chapter three shows, the fields where neural networks are applicable. The domain of applications of neural networks is extensive, and expanding, the degree of success has varied with the type of application.

Neural networks have been applied in solving a wide variety problem. Neural network can be applied for business, stock, medicine, industry, planning, monitoring, control, and many applications cannot be limited. However the most successful application is in medicine, such as classifying breast cancer cells.

Chapter four shows in details the important application of neural network in pattern recognition, almost any application will fit into pattern recognition area, there is some potential for neural network pattern recognition purposes, including special resources, and analysis.

Using neural network for pattern recognition gives the ability to decoding algorithm and predicting sequences, recognizing the chemical drawings, providing contents for OCR, recognizing voice mail … etc. Thus it is not possible to report on full extent of work going on.

CHAPTER ONE

BACKGROUND ON NEURAL NETWORKS

1.1 Overview

The information processing principles of biological neural networks have been applied to building a computer system for solving difficult problems whose solutions normally require human intelligence .The neural network approach has attracted wide attention and found a growing number of applications especially in the last decade. Neural networks process information in a similar way the human brain does. The network is composed of large number of highly interconnected processing elements (neuron) working in parallel to solve a specific problem. Neural networks learn by examples. They cannot be programmed to perform a specific task. The examples must be selected carefully otherwise useful time is wasted.

Neural networks solve problems by self-learning, and self-organisation. They derive their intelligence from the collective behaviour of simple computational mechanism at individual neurons.

Neural networks can organise, classify, convert, and learn patterns. A pattern is qualitative or quantitative description of an object or concept or event.

The human brain is set-up to perform in the way we use it mostly in the first two years after birth, senses, and understanding what is around us are all learnt by the brain then after words more complex perceptions around us are slowly learnt over years. This learning process is quite vital in the development of the neural network, and certainly different neural network from traditional computing systems.

The question now will we solve our daily problems easily by repalcing our brain by anueral computer, and ask it to solve these problems ? We may suggest that, and let's imagine that time when we ask the computer to solve our complicated problems, and show the result quickly.

1.2 Background on Neural Networks

Many tasks, which seem simple for us, such as reading a handwritten note or, recognising a face, are difficult for even the most advanced computer. In an effort to increase the computer's ability to perform such tasks, programmers began designing software to act more like the human brain, with its neurons and synaptic connections. Thus the field of "artificial neural networks" was born. Rather than employ the traditional method of one central processor (a Pentium) to carry out many instructions one at a time, the neural network software analyses data by passing it through several simulated processors, which are interconnected with synaptic-like "weights".

After training on a few dozen cases, the network begins to organise itself, and refines its own architecture to fit the data, much like a human brain "learns" from example. If there is any overall pattern to the data, or some consistent relationship between the inputs and result of each record, the network should be able to eventually create an internal mapping of weights that can accurately reproduce the expected output.

Once you realise how powerful this type of "reverse engineering" technology can be, you begin to understand why neural networks were once regarded as the best kept secret of large corporate, government, and academic researchers. Once only available to those with the training and the computing power, this advanced intelligence technique is now available to anyone using Microsoft Excel. Neural networks still require a lot of processing power, but they are now quite simple to use, and thanks to today's faster generation of desktop computers, there are fewer reasons to stick with the traditional statistical methods each year.

1.3 History and Principles of Neural Networks to 1960

1.3.1 The First Neural Network Theory

Alexander Bain (-1818 - 1903) of the United Kingdom presented the first neural network in his 1873 book entitled "Mind and Body The Theories of Their Relation" [1]. His work was no doubt inspired by the new findings of neuroanatomists who using the newly discovered neural stains of carmine. (By Gerlach in 1858), methylene blue (by Nisi in 1858), and haematoxylin (by Waldeyerin 1863 - stains axon fibers) were able for the first time to see the true extent of the interlocking fibres of the brain.

The 1853 Manual of Human Histology written by Rudolf Kolliker gives the state of knowledge before these stains [2], which states that there were various forms of nerve cells, and besides these there are a good many fine pale fibers. Like the processes of cells, only more extended of which nothing more can be said as to whether they are nerve tubes or are to be referred to as the processes of cells.

The ability to recall a specific memory requires that an association (grouping) first be made with some other memory, sensation, or motor action via some kind of neural growth.

"For every act of memory, every exercise of bodily aptitude, every habit recollection, train of ideas, there is a specific grouping, or coordination, of sensations and movements, by virtue of specific growths in cell junctions.

Yet Bain realised that if his theory were true every possible association or, grouping would have to be hardware into the brain. The number of fibres, and cells brought into action, before the grouping can converge upon some one set of cells definitely connected with an out-going motor arrangement, or with some other internal grouping, must be very great indeed. But for the vast number of fibres and cells, demonstrably present in the brain, the separate embodiment of every separate impression and idea would seem impractical.

1.3.2 First Neural Logic (1938 to 1943)

Yet no theory arose to challenge the holistic, reverberatory scheme of William James until 1938 when N.Rashevsky proposed that the brain could be organised around binary logic operations since action potentillas could be viewed as binary 1 (true) values, operations in analog terms and his formulation remained incomplete.

In 1943 Warren McCulloch, realised that the natural consequence of the standard neuron model's threshold in combination with binary action potentials produced another type of logic called threshold logic.

Threshold logic was expanded by adding more input lines so that the output of the summation node would become analog since this seemed to fit in more with the standard neuron model.

Yet by using more than two inputs the operations ceased to be logical, that is, they ceased to define the basic set of pattern classifiers used as building blocks in larger circuits to classify patterns based upon the pattern distribution. Instead these operations became parameter classifiers, which classified their input patterns based upon the one parameter of signal magnitude given by the summation of the binary inputs.

The very existence of an analog value for the first time allowed these operations to become adaptive using multiplication factors called weights. Exactly how that adaptability was implemented defines the many different neural networks described below. Yet modifications to a single parameter can never fully characterise the distribution (the pattern) of an operation's input. Some ambiguity is always the result yet at the time this ambiguity was hailed as a great achievement of neural networks. For many saw it as meaning that neural networks could work with partial knowledge even though this ambiguity could never be precisely defined as it is in a classification hierarchy

So the achievement of adaptability in the neural networks of this time came at the cost of loosing the decision-making resolution of logic. By 1963 this split theoretical neuro scientists into the separate groupings of neural network researchers and artificial intelligence researchers.

The 1963 paper by R.O. Winder [3] seems to have been the last major paper, which still attempted to combine the weight based, and mathematical neural network approach with the logical artificial intelligence approach. But its approach was to use mathematical techniques to find the desired weights of a network instead of learning them. The result was that the neural network researchers continued to develop parameter based classifier circuits while the artificial intelligence researchers expanded and abstracted the binary logic approach to form propositional and prepositional logics which eventually resulted in the development of the programming language called LISP. Consequently, because of the precision of logic the field of artificial intelligence enjoyed great initial success. Yet its high level algorithmic and sequential approach eventually proved to have serious limitations not only in terms of adaptability but also in terms of general creative design since high level design tools, and languages impose as many limitations as conveniences.

1.3.3 The First Randomly Connected Reverberatory Networks (1954)

Neural network research really only became possible at the dawn of the computer age when ideas could be validated by simulation on various types of electronic calculators. The impetus for these simulations was provided by McGill University in Canada, who in 1949 proposed this unidirectional variation of the William James learning rule: "Let us assume then that the persistence of repetition of a reverberatory activity (or trace) tends to induce lasting cellular changes that add to its stability" [4].

The assumption can be precisely stated as: "When an axon of cell A is near enough to excite cell B, and repeatedly, or persistently takes part in firing it.

Some growth process, or metabolic change takes place on one, or both cells so that A's efficiency as one of the cells firing B is increased". The first such Hebbian inspired network was simulated by Farley, and Clark in 1954 on an early digital computer.

Yet in order to get this network to work Farley and Clark had to modify Hebb's learning rule. This new rule required that the activity level of each network line be examined at each instant in time so that if the line value changed in the desired direction then all of its nodal input weights would be incremented (increased). If the change was in the wrong direction then the nodal input weights would be decremented (decreased). With this rule the network was able to successfully discriminate between two widely differing patterns as long as they were presented alternately .The major problem with this type of network is its lack of pattern discrimination resolution combined with the high number of neurons needed to make that discrimination. Also the cells in the network were not self-assembled as was thought to be necessary at the time but were assigned to assemblies (the quadrants) by the experimenters [4].

1.4 History and Principles of Neural Networks from 1960 to Present

In 1962 Frank Rosenblatt published a book which combined the concepts of his original perceptron with those of ADALINE to come up with the classic perceptron contrast to ADALINE, perceptrons are based on repulsive learning in which only the weights on the non-active lines are changed in response to an error. In other words the weights change only in response to a misclassification. Thus the weight values are not pulled towards some defined goal but are pushed away from non-goals. Consequently each subcircuit can represent a whole class of patterns.

The adaptive multiplication factors (weights) are now placed before the summation node like ADALINE instead of after the node as in the original perceptron. In addition all convergent subcircuits now share a common set of inputs instead of having randomly connected inputs (although the initial values of the weights may be randomised which would effectively accomplish the same thing). These changes allowed the input pattern to dispense with the binary line signal requirement in favour of more realistic analog signals, which could represent the frequency of an action potential pulse or the ionic charge on a neuron. Yet, in order for patterns to be reliably discriminated by perceptrons the pattern inputs had to be normalised, that is the numbers in each pattern had to add up to the same value - usually one. Using analog values (and thus analog equations) also required that the binary threshold be replaced with a subtractive threshold.

The 1960's also saw the growth of artificial intelligence techniques based mostly on net search techniques and higher order logic languages known as propositional and predicate calculus. The aim of artificial intelligence (A.I.) researchers was to simulate intelligent processes at a level more abstracts than that of the neural level and they were having good success at the time. They could see what was required of any intelligent machine so when the supporters of perceptrons began to oversell its potential one of A.I.'s founders, Marvin Minsky (who had started out with neural networks), with Seymour Papert were inspired in 1969 to write a book describing the Perceptron's inherent limitations. Since the perceptron was the most sophisticated neural network idea at the time that book essentially ended neural network research in the United States for a period of time.

In 1975, inspired by the self-organisation ability of the brain, Kuniko Fukushima from Japan introduced the Cognitron network as an extension of the original perceptron. Like the Original Perceptron the Cognitron is a pattern regularity detector meaning it is able to learn patterns without some mechanism (a teacher) to indicate the success or non-success of a pattern match. Unlike the original Perceptron the Cognitron is better able to handle (but not perfectly) the pattern subset problem in which one pattern is completely contained within the other. It does this by using a special inhibitory input to the convergent subcircuit node, which tends to counteract the effects of larger patterns. Also unlike the original Perceptron the Cognitron can discriminate to some degree between analog patterns although binary patterns are usually presented to the first layer.

In 1982, John Hopfield revived interest in neural networks in the United States with the introduction of a new type of reverberatory network. It differed from the earlier versions by using bi-directional lines (equivalent to two reciprocal unidirectional lines) between summation nodes instead of unidirectional lines and emphasised individual cells (nodes) instead of cell assemblies. The summation nodes have a threshold of zero and only produce an output a one if that threshold is met or exceeded. The weight values between the nodes can be any number between -1 and 1 so both exitory and inhibitory operations are represented. The general learning idea behind the network is that the weights between the active nodes (those producing an output of 1) will increment while those between all other nodes will decrease. This is usually accomplished by normalising all the weights. In the original paper by Hopfield all the weights added up to zero [4].

So in order to learn a correlation (association) between some input and output pattern a separate training session must be held in which the input pattern is presented along with the desired output pattern. In the original paper the weights were assigned initial values at random. The weights between active nodes are then incremented by some amount and the normalisation process then decrements the remaining weights, the process is repeated for all patterns until the weights stop changing (or nearly so).

1.5 What is a Neural Network ?

There are many definitions explain the meaning of the neural network, and can be summarised as follows:

Neural network: is a system composed of many simple processing elements operating in parallel whose function is determined by network structure, connection strength, and processing performed at computing elements or nodes.

Neural network: is a buzzword. Why? They are a very powerful tool in non - linear statistical analysis .as such they have found their way into many fields –control theory, natural language processing, image processing, process modelling, and are strongly supported by industry.

Neural network: Neural networks are different paradigm for computing:

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

Neural network: Neural networks are forms of multiprocessor computer system,

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

Neural network: It is a concept of processing data based on the way neurons in the brain process information, and communicates with each other [5].

Neural network: It is a system of programs, and data structures that approximates the operation of the human brain. A neural network usually involves a large number of processors operating in parallel, each with own small sphere of knowledge and access to data in its local memory. Typically, anural network is initially "trained" or fed large amount of data and rules about data relationships (for example, "A grandfather is older than a person's father"). A program can then tell the network how to behave in response to an external stimulus.

Neural network: Is a powerful computational tools that can be used for classification, pattern recognition, empirical modeling and for many other tasks. Neural network (or artificial neural networks – a longer but more correct definition) can be "trained " to provide the right output (binary, fuzzy, quantitative) if enough input-output patterns are available and if these patterns effectively describe the system that is to be modelled.

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

Neural network: Is a statistical analysis tool that is they let us build behaviour models starting from a collection of examples (defined by a series of numeric or textual) of this behaviour. The neural net, ignorant at the start, will, through a learning process, become a model of the dependencies between the descriptive variables and the behaviour to be explained.

1.6 What is Artificial Neural Network ?

Also referred as connectionst architectures, parallel distributed processing, and neuromorphic systems, an artificial neural network (ANN) is an information-processing paradigm inspired by the way the densely interconnected, parallel structure of the mammalian brain processes information. Artificial neural networks are collections of mathematical models that emulate some of the observed properties of biological nervous systems and draw on the analogies of adaptive biological learning.

The key element of the ANN paradigm is the novel structure of the information processing systems. It id composed of a large number of highly interconnected processing elements that are analogous to neurons, and are tied together with weighted connections that are analogous to synapses.

Learning in biological systems involves *adjustments to the synaptic connections* that exist between the neurons. This is true of ANNs as well. Learning typically occurs by examples through training, or exposure to a truthed set of input/output data where the training algorithm iteratively adjusts the connection weights (synapses). These connection weights store the knowledge necessary to solve specific problems.

Although the ANNs have been around since the late 1950's that algorithm became sophisticated enough for general applications. Today ANNs are being applied to an increasing number of real-world problems of considerable complexity. They are good pattern recognition engines and, robust classifier, with the ability to generalise in making decisions about imprecise input data. They offer ideal solutions to a variety of classification problems such as, speech, characters, and signal recognition, as well as functional prediction and system modelling where physical process are not understood or highly complex. ANNs may also applied to control problems, where the input variables re measurements used to drive an output actuator, and the network learns the control function. The advantage of ANNs lies in their resilience against distortions in the input data, and their capability of learning. They are often good at solving problems that are too complex for conventional technologies.

There are multitudes of different types of ANNs. Some of the more popular include the multilayer perceptron, which is generally trained with the back propagation of error algorithm, learning vector quantisation, radial basis function, Hopfield, and Kohonen, to name a few. Some ANNs are classified as feedforward while others are recurrent (i.e. implement feedback) depending on how data is processed through the network. Another way of classifying ANN types is by their method of learning, as some ANNs employ supervised training while others are referred as unsupervised. Supervised training is analogous to a student guided by an instructor. Unsupervised algorithms essentially perform clustering of the data into similar groups based on the measured attributes.

1.7 How old are Neural Networks?

The idea of the neural networks has been around since the 1940s but only in the late 1980s were they advanced enough to prove useful in many areas such as computer vision, control and speech recognition. After that, interest exploded and neural networks were hailed as the miracle cure to all problems. Neural networks were quickly applied to financial forecasting with more or less success. Hard lessons were learned in those days, that is not enough to just 'throw some data at a neural network and, it will work '.

1.8 What is a Neuron?

A neuron is a building block of a neural network. It is very loosely based on, The brains nerve cell. Neurons will receive inputs via weighted links from other neurons. This input will be processed according to the neuron activation function. Signals are based on to other neurons.

There are three types of neurons within neural networks. Input neurons receive encoded information from the external environment. Output neurons send signals out to external environment in the form of encoded answer to the problem presented in the input. Hidden neurons allow intermediate calculation between inputs and outputs.

1.9 Why are Neural Networks Useful?

Neural networks are unlike artificial intelligence software in that they are trained to learn relationships in the data they have been given. Just like a child learns the difference between a chair, and a table by being shown examples, a neural network learns by being given a training set. Due to its complex, non-linear structure, the neural network can find relationships in data that humans are unable to do.

1.10 Why Neural Network Doesn't Work All The Times?

Neural networks can only learn if the training set consists of good examples. The old saying of 'garbage in garbage out' is doubly true fore neural networks. Great care should be taken to present decorrelated inputs, remove outliers in the data, and use as much prior knowledge to find relevant inputs as possible. Care must also be taken that the training set is representative, a neural network cannot learn from just a few examples.

1.11 What are the A advantages of Neural Networks?

1. Neural networks can be retrained using additional input variables.
2. Once trained, they are very fast.
3. Due to increased accuracy, results in cost saving.
4. They deal with the non-linearity in the world in which we live.
5. They handle noisy or missing data.
6. They create their own relationships amongst information – no equation!
7. They provide general solutions with good predictive accuracy.

1.12 What are the Disadvantages of the Neural Networks?

1. No set rules for network selection.
2. Needs expertise in training the network.

1.13 Where are the Neural Networks Applicable ?

Neural network cannot do anything that cannot be done using traditional computing techniques, but they can do some things, which would otherwise be very difficult.

In particular, they can form a model from their training data (or possibly input data) alone.

This is particularly useful with sensory data, or with data from a complex (e.g. chemical, manufacturing, or commercial) process. There may be an algorithm, but it is known, or has too many variables. It is easier to let the network learn from examples.

Neural networks are being used in:

1. **Investment analysis:** To attempt to predict the movement of stock currencies, from previous data. There, they are replacing earlier simpler linear models.

2. **Signature analysis:** As mechanism for comparing signature made (e.g. in a bank) with those stored. This is one of the first large-scale applications of neural network in the USA, and is also one of the first to use a neural network.
3. **Process control:** There are clearly applications to be made here: most processes cannot be determined as computable algorithms. Newcastle University Chemical Engineering Department is working with industrial partners in this area.
4. **Monitoring:** Networks have been used to monitor the state of aircraft engines, by monitoring vibration levels and sound, early warning of engine problems can be given.
5. **Marketing:** Networks have been used to improve marketing mailshots. One technique is to run a test mailshots, and look at the pattern of returns from this. The idea is to find a predictive mapping from the data known about the clients to how they have responded. This mapping is then used to direct further mail shots.
6. **Pen PC's:** PC's where one can write on a tablet, and writing will be recognised, and translated into (ASCII) text.
7. **Speech and vision recognition systems:** Not new application but neural networks are becoming increasingly part of such systems. They are used as a system component, in conjunction with traditional computers.
8. **Whites goods and toys:** As neural network chips become available, the possibility of simple cheap systems, which have learned to recognise simple entities (e.g. walls looming, or simple commands like Go, or Stop), may lead to their incorporation in toys and washing machines etc. Already the Japanese are using a related technology, fuzzy logic, in this way. There is considerable interest in the combination of fuzzy and neural technologies.
9. **Forecasting:** Future sales, production requirements, market performance, economic indicators, energy requirements, medical outcomes, chemical reaction products, weather, crop forecast, environmental risk, horse races, jury panels.

1.14 Summary

We cannot limit the meaning of neural network over some lines. But the important thing is that we should know at least one definition of a neural network. As we have explained in the sections above a neural network is a concept of processing data based on the way neurons in the brain process information. We also have considered the history beginning from 1960 up to present, we have seen the development, and progress of the architecture of neural network, also we have explained the meaning of artificial neural network. The meaning of neuron was our important definition as it is the building block of a neural network. We also have given many examples where a neural network is applicable e.g. speech and vision recognition, and forecasting... finally we have described the advantages and disadvantages of neural network, as one important advantage was that it can handle noisy or missing data.

CHAPTER TWO

STRUCTURES OF NEURAL NEWORKS

2.1 Overview

Neural Networks have been hailed as the greatest technological advance since the transistor. They are so named because their design is based on the neural structure of the brain on which scientists (Neurobiologists) have been doing intensive researches to understand its biological structure and behaviour. There are two types of neural networks, the biological neural networks and the artificial neural networks, which are computer (software & hardware) simulations of the biological ones. This document is an introduction to the technology of Neural Networks along with a brief description of the biological side of it.

2.2 Biological Neural Networks

Neural networks architectures are motivated by models of our own brains, and nerve cell. Although the knowledge of the brain is limited, we do have much detailed anatomical and physiological information. The basic anatomy of an individual nerve cell (also known as the neuron) is known, and the most important biochemical reactions that govern its activities have been identified.

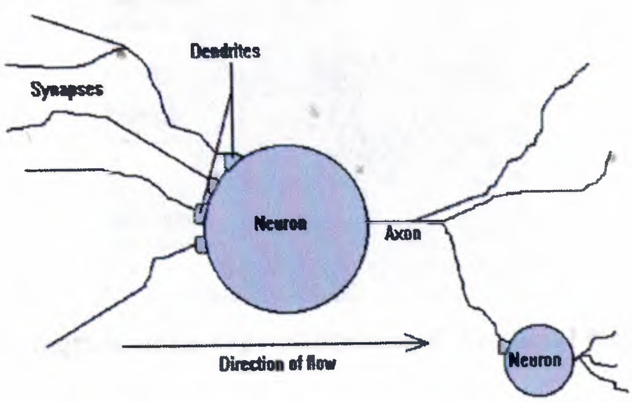

Figure 2.1. A simple neuron cell

The biological brain is an incredibly complex system of more than a 100 billion neurons of different types (not all) highly interconnected with each other via synapses of which there are more than a 150 billion. There is a set of synapses coming into each neuron which communicate with it through it's private dendrites, and each neuron also have an axon out of which it delivers it's messages to other neurons. It is also known that the human brain performs an average of 100 operations per second. Action potentillas are fired from each neuron to others (depending on the task the brain is performing), which are electric pulses whose intensity level varies [6].

In comparison Artificial Neural Nets (ANN), like their biological equivalents, consist of processing elements called neurons or units, and connections between them called synapses or just connections. Associated with each connection is a weight that simulates the synaptic gap. Instead of pulse trains, most ANNs use analogue values as a means of communication. Floating point numbers in software simulations usually represents these.

There are three types of neuron, input, output, and hidden. Input, and output neurons form the nodes at which data enters or leaves the network; hidden neurons, as their name implies, are internal to the network.

A modern ANN, then, consists of these elements joined by connections, and is normally represented by fig.2.2 similar to that shown below.

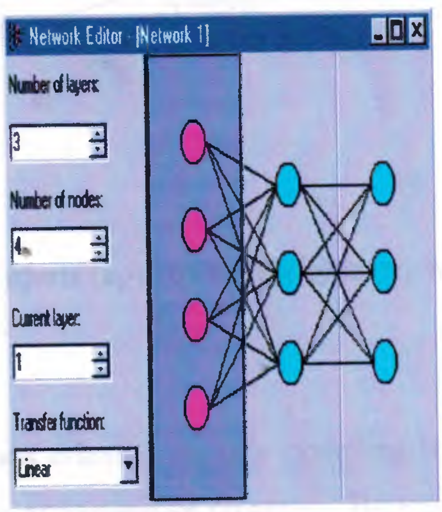

Figure 2.2. A diagrammatic representation of Artificial Neural Network

2.3 First Revolution

In contrast, the first adaptive systems were built in parallel with the invention, and early development of the digital computer, and thus, simulation on digital computers was not available as a means of exploring adaptive systems.

McCulloch and Pitts did early work in 1943 creating simple logic circuits composed of interconnected neuronlike elements. However, the first approach to biological neurons was the "Perceptron", invented by Frank Rosenblatt in 1957. The perceptron was actually an entire class of architectures, composed of processing units that transmitted signals and adapted their interconnection weights. Fig.2.3 below shows a perceptron with 5 inputs [7].

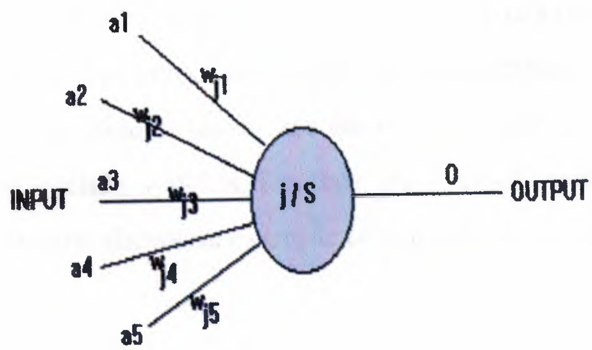

Figure 2.3. Showing Inputs (a_1-a_5), and Weights (w_1-w_5) Summed at Neuron Input

Rosenblatt's research was oriented towards modelling the brain in an attempt to understand memory, learning, and cognitive processes. However, while Rosenblatt was interested in the properties of the brain, other scientists and engineers were eager to characterise the capabilities of the perceptron, and to experiment with possible applications. And although the research did not lead to viable applications at the time, due to lack of breadth and poor computer facilities, the perceptron nevertheless provided an architecture that was eventually extended to the neural network learning systems today.

2.4 The Perceptron

This is a single processing unit (neuron) with incoming input lines, and one output line. Works as a Binary Decision Neuron (BDN).

I.e. each perceptron possesses a threshold value and every time an input pattern is given to the perceptron, it computes a net weighted value out of this pattern, then compares it to the threshold. If the threshold is bigger than the net input, then a zero value is given at the output; otherwise, a one is given as an output.

2.5 Second Revolution - New neural Networks Architectures

The new generation neural networks, which appeared in the early eighties, were much more powerful than the perceptrons, and much more brain like systems. Most of these neural networks architecture consist of, an input layer, an output layer, which are the only two layers that are in contact with other systems outside the network, and at least one hidden layer. The hidden layers are named so because of their property of being invisible from the outside systems. I.e. they are internal layers used only by the neural network. Fig.2.4 below shows an example of neural network architecture.

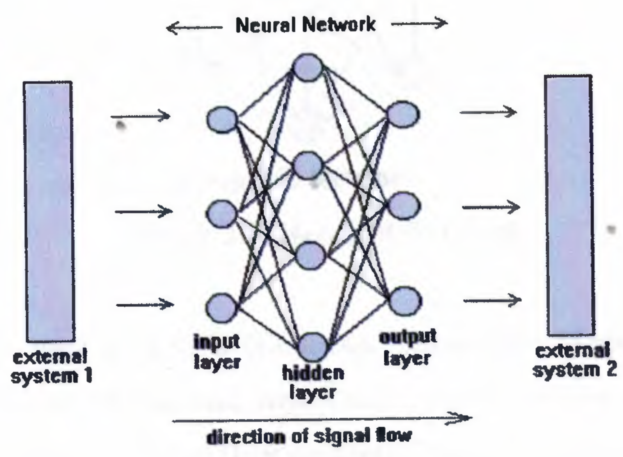

Figure 2.4. Classical Multi-Layer Perceptron (MLP) Feed Forward Network

One type of this generation's neural networks architectures drawn above has three layers an input layer, an output layer and a hidden layer. Only the input and output layers communicate directly with outside systems. The input layer is used to present signals to the network (one signal for each neuron), and no processing, or change occurs to these signals at this level. The output layer is almost similar to the input layer apart from the fact that it is used to detect the neural network's outputs, which basically are the actual outputs of the neurons on this layer. The hidden layer is the feature detector. I.e., it responds to particular features that may appear in the input layer.

Neural networks which have an architecture similar to the one described above are known as the "Feed Forward Neural Nets" because of their property of forwarding signals in one direction without returning nor feeding them back. As the picture above shows, neurons on a same layer are not connected to each other, they have connections coming into them from the neuron in the layer just below and connections going from them to the neurons on the next layer above [8].

Another architecture which appeared around that time but which is less powerful is the "Feedback (Recurrent)" neural networks. The recurrent neural nets have a complete connectivity. There is neither layers layout nor difference between neurons. The most well known neural network with such architecture is the "Hopfield" network. These neural networks have very limited applications.

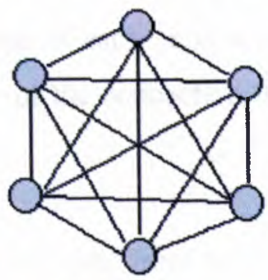

Figure 2.5. Hopfield Network

They are mostly used as CAMs (Contents Addressable Memories) because of their ability to produce a set of minimas, which can be used as memory cells, and items are stored at addresses, that depend on their contents. The style of computation this type of neural networks have is known as relaxation. I.e., they process an iterative convergence to some fixed point, which is nearer to the starting point. When a trained Hopfield network is queried, the query pattern is given to the net, by assigning each single signal of this pattern to a single neuron. After that the network will switch some neurons ON and others OFF to bring the initial state of the net to another which it has trained on and which it thinks is closer to it. This is a typical situation in pattern recognition [9].

2.6 Supervised Neural Networks

2.6.1 How Supervised Learning Works (Back-Propagation)

This section contains a full mathematical description of how supervised neural networks learn (train). The most frequently used and effective supervised learning algorithm known in the world of neural networks is the "Back-Error Propagation Algorithm" or Back-Prop for short. The type of neural networks this learning algorithm requires is "the feed forward neural networks". It is for this reason they are also known as "back-propagation neural networks. Being a supervised learning algorithm, the back-error propagation relies on a teacher, which is a set of example pairs of patterns. The basic idea of the way this algorithm works is the following [10].

First a pair from the training data set is chosen randomly. The input pattern of the pair is given to the network at the input layer by assigning each signal of the pattern to one neuron on this layer. Then, the network passes these signals forward to the neurons on the next layer (hidden layer). But, how is this done?

For each neuron on the hidden layer, a **Net Input** value is computed. By doing the sum over the products of the output of each neuron on the input layer. Which is the original signal itself by the weight of the connection that connects it to the neuron on the hidden layer in question. I.e.,

$$\mathbf{NetL_{pi} = å\ O\ (L\text{-}1)}_{pjWij}$$

p : is the index of the pair of patterns chosen from the examples set.

$\mathbf{NetL_{pi}}$: is the net input of neuron **i** on layer **L** corresponding to pattern p.

$\mathbf{O\ (L\text{-}1)}_{pi}$: is the output of neuron **j** on the layer just below **L**. I.e., **(L-1)** corresponding to pattern **p**.

$\mathbf{W_{ij}}$: is the weight of the connection from neuron **j** to neuron **i**.

When all the neurons on this layer have received a Net Input, the next step for each of these neurons is to compute, from it's Net Input, an **activation** value which is also considered as it's output. This process is done using a transfer function, usually the **sigmoid** function in the following way:

$$\mathbf{OL_{pi} = 1 / (1 + e^{-Net}L_{pi})}$$

Then, these outputs are passed forward to the next layer and the same processes of computing net inputs and activation are done, until the output layer of the neural network is reached. The output values of the neurons on the output layer are taken as one pattern of signals, which is considered as the **actual** output pattern of the network.

The actual output pattern that the network produces for each input pattern is compared to the target output pattern it should have produced which is simply the second element of the example pair chosen randomly at the beginning of the whole process. An error value is computed using the actual and target patterns as follows:

$$\mathbf{E_p = å\ (O_{pi} - T_{pi})^2}$$

Where:

$\mathbf{E_p}$: is the error value that corresponds to example pair **p**.

$\mathbf{O_{pi}}$: is the output value of neuron **i** on the output layer of the network.

$\mathbf{T_{pi}}$: is the i'th signal value on the target output pattern of example pair **p**.

If the value of this error is zero, there will be no need to make any changes in the connectivity state. However, if the error value is not zero, some changes are to be made in the weights of the connections in the network reduce this error. The way this is done is as follows.

We should bear in mind that this process, as the title of the algorithm actually states, involves sweeping the error backwards through the network and at each layer (level) the relevant changes are made to the weights of the connections, which we will discuss in the following.

Each weight is either increased by some fraction or decreased. The mathematical formula used by this algorithm is known as the **Delta Rule**. Which is:

$$\mathbf{D\ {}_pW_{ij} = hd_{Lpi}O_{Lpi}}$$

Where:

$\mathbf{D\ {}_pW_{ij}}$: is the amount by which the weight $\mathbf{W_{ij}}$ should change correspondingly to training pattern pair p

h : is the learning rate

$\mathbf{d_{Lpi}}$: is the error on the output of unit **i** on layer **L** for pattern pair **p**. The computation of it's value depends on the type of the neuron in question.

The way the error at the output of a neuron is computed depends on the type of the neuron. So if it's an output neuron then the error on it is:

$$\mathbf{d\ {}_{Lpi} = (T_{pi} - O_{Lpi})OL_{pi}(1 - O_{Lpi}).}$$

However if it's a hidden neuron then the error value on it is:

$d_{Lpi} = O_{Lpi}(1 - O_{Lpi})Sd(L+1)_{pk}W_{ki}$.

Where:

$d(L+1)_{pk}$: is the error value of neuron **k** on the layer just above layer **L**. That is layer **(L+1)**

W_{ki}: is the weight of the connection going from the neuron in question **i** to neuron k on the layer just above.

The learning rate is a value that must be chosen between 0, and 0.9. It determines the size of the step by which the neural network system moves towards an optimal state. The actual idea behind the back-error propagation algorithm is to slide along the error surface performing a gradient descent in search of, ideally, what is known a global minima, i.e. a state of the network, where the error on it's output patterns in optimal (minium).

Fig.2.6 below shows a typical example of an error surface of a neural network system on which the state of this system should slide in search of global minima.

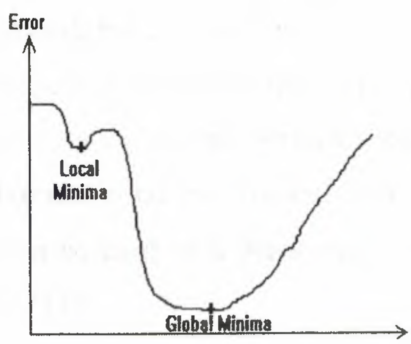

Figure 2.6. Example of Local Minima

Fig.2.6 above also shows an example of what is known as local minima, which is simply an area on the error surface where the error of the system drops down, but it is not a good solution to the problem.

Choosing a value for the learning rate is very delicate, because, if it's assigned a large value then local minimas can easily be avoided by just jumping over them, but this might end the system up in oscillation. I.e., jumping forward and backward over global minima without ever getting there. However if the learning rate is given a small value, then may be global minimas cannot be missed, if there are any around, but the system is more likely to be trapped in a local minima. For this reason actually, a new variable has been introduced, known as the **Momentum**, whose value should be in the range 0 to 0.9 as well. The momentum times the old correction to the weights is added all the time a new correction is being proceeded. This way, the learning rate value can take a large value and the risk to end up in an oscillating state is minimised. The final mathematical formula used by the back-error propagation algorithm to update the connection weights in a feed forward neural network is:

$\mathbf{NEWD_pW_{ij} = hd_{Lpi}O_{Lpi} + a\ OLDD_pW_{ij}}$

Where:

$\mathbf{NEWD_pW_{ij}}$: is the new weight correction value of W_{ij} concerning pattern p

$\mathbf{OLDD_pW_{ij}}$: is the old weight correction value of W_{ij} concerning pattern p

a : is the momentum.

This whole process is done for each and every example pair and for many epochs. Once a neural network has been trained to do a certain task, it should then be validated. The process of validation is in other words a process of checking its performance. This is done by providing a set of pairs of input/output patterns which is similar to the training set used to teach the network but different in contents. With this set of data, we give the input patterns to the network and observe the output produced then compare it to the target output. A judgement on the overall performance of the network, whether some more training is required or not, is taken there and then. Once the network is fully trained and validated, it can then be used as a black box system that one may query using it's input and output layers [11].

2.7 Unsupervised Neural Networks

2.7.1 What is Unsupervised Learning?

Unsupervised learning is a process when the network is able to discover statistical regularities in its input space, and automatically develops different modes of behavior to represent different classes of inputs (in practical applications some 'labelling' is required after training, since it is not known at the outset, which mode of behaviour will be associated with a given input class). Kohonen's self-organising (topographic) map neural networks use this type of learning.

2.7.2 Kohonen Self Organising Map (SOM) - Unsupervised Neural Network

A very effective and frequently used un-supervised neural network architecture is the "Kohonen" neural network. These networks have only two layers, a standard input layer and an output layer known as the "Competitive (Kohonen)" layer (the reasons for which it is called so will be discussed later in a following paragraph).

Each input neuron is connected to each and every neuron on the competitive layer which are organised as a two dimensional grid. The picture bellow shows a typical example of a Kohonen network with 2 inputs and 25 neurons on the competitive layer.

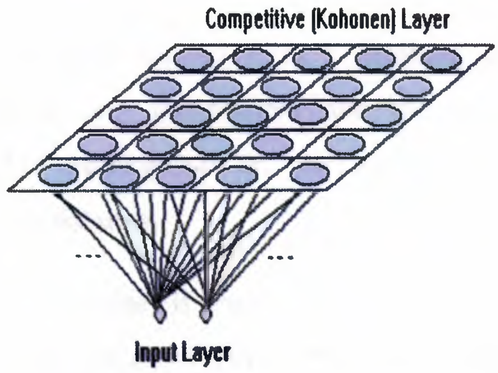

Figure 2.7. A Kohonen Self Organising Grid - 2 Dimensional Output Layer

The input layer in a Kohonen network has the same function as the input layer described in the feed forward networks. However, the neurons on the output layer have a totally different property, they can actually find the organisation of relationships among input patterns which are classified by the competitive neurons that they activate. Kohonen networks are known to be self-organising feature maps (more details will be given). I.e., they can organise a topological map from a random starting point and the resulting map shows the natural relationships among the patterns that are given to them.

Topological mapping of sensory and motor phenomena exist on the surface of the brain. It is important to keep in mind, however, that the brain mechanisms are different from the paradigm described here. The detailed structure of the brain is different, and input patterns are represented differently in biological systems. Furthermore, biological neural systems have a much more complex interconnection topology. However, the basic idea of having a neural network organise a topological map is illustrated effectively with the Kohonen neural networks [12].

2.7.3 What Can Kohonen Network Be Used For?

Kohonen networks can be used for various tasks:

- **Classification** (where none exists)
- **Marketing**: A campaign is being run to increase sales of a product. It is intended to do a targeted mailshot to individuals most likely to be interested in the product.

There is information available from a consumer survey, which includes information about current purchasers of the product. An unsupervised net is trained using information from the consumer survey. The groups formed are examined and all individuals in the groups rich in existing purchasers are used as targets for the mailshot.

- **Data Cleaning**: Data being extracted from a database is being analysed but unfortunately the data is too inconsistent for the analysis technique. An unsupervised net is used to create groups and the analysis is based on the individual groups instead of the entire database.

- **Anomaly Discovery**: A machine is being monitored for unusual behaviour. However there is no clear definition of unusual behaviour except that it is behaviour not found to date. An unsupervised net is trained using the normal readings for the machine. This forms groups, which represent the normal running of the machine. New data is then checked to see that it falls within one of the existing groups. An alarm is raised when a reading falls outside all existing groups.

- **Combined with supervised neural nets**: A supervised neural net requires targets and in many cases this is not available from the existing data. An unsupervised net can be used to create a target. An unsupervised net is trained and each possible group is given an arbitrary name. Data is passed through the unsupervised net and the group that a row is a member of becomes the target for the supervised net. This technique is often used to discover more information about the groups by, for example, using Neufuzzy, a neuro-fuzzy system, to create a set of rules that define the characteristics of the group.

2.7.4 How a Kohonen Network Works?

The most famous unsupervised learning algorithm known in neural networks is the Kohonen algorithm. The way this algorithm works is completely different from the way back-error propagation works.

The network is presented with a set of training input patterns without giving any target output pattern for any of the inputs. One of the patterns is chosen randomly. Each neuron in the input layer of the network takes on the value of the corresponding entry in the input pattern. Then, a distance value (also known as the **Euclidean Distance**) is computed for each of the neurons on the competitive layer. The competitive neuron with the smallest Euclidean distance is known as the **winning node**. The way this first step is proceeded is as follows:

Say the chosen input pattern from the data set is:

$\mathbf{I = (I_1, I_2, I_3, ..., I_n)}$

I.e., There are '**n**' input neurons in the input layer.

Now suppose there are '**m**' neurons in the competitive layer, where we call **Ui** the i'th neuron. So the whole of the competitive layer will be:

$\mathbf{U = (U_1, U_2, U_3, ..., U_m)}$

For each competitive neuron $\mathbf{U_i}$ there is a set of n incoming connections from the n input neurons on the input layer. Each connection has a weight value **W**, so for any neuron $\mathbf{U_i}$ on the competitive layer the set of it's incoming connections weights is:

$\mathbf{W_i = (W_{i1}, W_{i2}, W_{i3}, ..., W_{in})}$

The Euclidean distance value $\mathbf{D_i}$ of a neuron $\mathbf{U_i}$ in the competitive layer whenever an input pattern **I** is presented at the input layer is the following:

$$D_i = \sqrt{\sum_{j=1}^{n}(I_j - W_{ij})^2}$$

The competitive unit with the lowest distance at this stage is the closest to the current input pattern therefore it is its representative. Let's call it $\mathbf{U_c}$. After the winning node is identified, the next step is to identify the neighbourhood around it, which is simply the set of competitive units, which are close to that winning node. Fig2.8 bellow shows an example of a neighbourhood around a winning node, which is just the set of neurons that are within the square that is centred on the winning node $\mathbf{U_c}$.

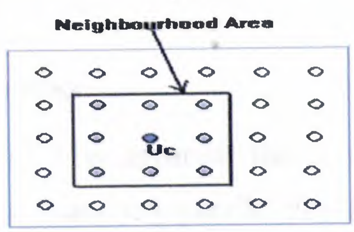

Figure 2.8. A 6x5 Kohonen Grid showing the size of neighbourhood influence around node Uc

solve complex problems. The use of neural networks offers the following properties and capabilities:

1. **Nonlinearity:** A neuron is basically a nonlinear device. Consequently, a neural network, made up of an interconnection of neurons, is itself nonlinear. Moreover, the nonlinearity is a special kind os sense that it is distributed throughout the network. Nonlinearity is a highly important property, particularly if the underlying physical mechanism responsible for the generation of an input signal (e.g...speech signal) is inherently nonlinear.
2. **Input-output Mapping:** A popular paradigm of learning called supervised learning involves the modification of the synaptic weights of a neural network by applying a set of labelled training samples or task samples. Each example consists of a unique input signal, and the corresponding desired response. The network is presented an example picked at random from the set, and the synaptic weights (free parameters) of the network are modified. So as to minimise the difference between the desired response, and the actual response of the network produced by the input signal in accordance with an appropriate statistical criterion. The training of the network is repeated for many examples in the set statistical criterion. The training of the network is repeated for many examples in the set until the network reaches a steady state.
3. **Adaptivity:** Neural networks have a built-in capability to adapt their synaptic weights to changes in the surrounding environment. In particular a neural network trained to operate in a specific environment can be easily retrained to deal with minor changes in the operating environmental conditions. Moreover, when it is operating in a nonstationary environment (i.e.. one whose statistics change with time), a neural network can be designed to change its synaptic weights in real time. The natural architecture of a neural network for pattern classification, signal processing, and control application, coupled with the adaptive capability of the network, make it an ideal tool for use in adaptive pattern classification, adaptive signal processing, and adaptive control. As a general rule, it may be said that the more adaptive system is stable, the more robust its performance will likely be when the system is required to operate in a nonstationary environment.
4. **Evidential Response:** In the context of pattern classification, a neural network can be designed to provide information not only about which

particular pattern to select, but also about the confidence in the decision made. This latter information may be used to reject ambiguous pattern, should they arise, and thereby improve the classification performance of the network.

5. **Fault Tolerance:** A neural network, implemented in hardware from, has the potential to be inherently fault tolerant in these in the sense that its performance is degraded gracefully under adverse operating conditions (Bolt, 1992). For example, if a neuron or its connecting links are damaged, recall of a stored pattern is impaired in quality. However, owing to the distributed nature of information in the network, the damage has to be extensive before the overall response of the network is degraded seriously.
6. **Uniformity of Analysis and Design:** Basically, neural networks enjoy universality as information processors. We say this in sense that the same notation is used in all the domains involving the application of neural networks. This feature manifests itself in different ways.

- Neurons, in one form or another, represent an ingredient common to all neural networks.
- This commonality makes it possible to share theories, and learning algorithm in different applications of neural network.

2.9 Summary

The most important thing after this discussion is that we should know that there is no limit for the structure of the neural network. In this chapter We have described some of the most frequently used neural networks with their essential features, learning processes, and how they are used once trained, with a brief look at the biological neural networks from which the idea of this technology has been derived.

The fields of applications of neural networks are limitless. In many situations, artificial neural networks are best solution, especially when it comes to dealing with non-linear and complex problems.

Also we have discussed the classification of neural network according to its type of learning, we have seen two types of learning, the supervised and unsupervised neural network and under each type we have seen many algorithms such as a Kohonen algorithm. Finally we have explained the benefits of the neural network.

CHAPTER THREE

APPLICATIONS OF NEURAL NETWORKS

3.1 Overview

In parallel with development of theories, and architecture for neural networks, the scopes for applications are broadening at a rapid pace. It is impossible for us even to attempt to cover every application.

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

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

3.2 Business Applications

3.2.1 Credit Scoring with Neural Network Software

According to research conducted by Herbert L. Jensen, Ph.D., an Ernst & Young Research Fellow at California State University Fullerton, "building a neural network capable of analyzing the credit worthiness of loan applicants is quite practical and can be done quite easily [13]."

The credit scoring neural network was trained on no more than 100 loan applications yet achieved a 75-80% success rate. One day's work by an operator familiar with the BrainMaker software package was required to build, train, and test the credit scoring neural network. Except for showing a greater bias towards approving weak loan applications, the neural network's loan classification rate was identical to that achieved using a commercial credit-scoring scheme.

The input data for the credit scoring with Brainmaker neural network software study consisted of information typically found on loan applications. The outcomes of those loans were classified as either delinquent, charged-off, or paid-off. The actual outputs from the network were 0 to 1 ratings for the three alternatives.

Once the network was built, it was subjected to two training trials. In the first trial, the data was arranged in random order, and the first 75 applications were used to train the network. The remaining 50 applications were then evaluated using the trained network. The network misclassified 10 of the 50 applications in the sample for an 80% success rate. In short, the network favored approving loan applications. More traditional, and much more costly, credit scoring method used by 82% of all banks, resulted in a 74% success rate. The credit scoring method proved to be more conservative than the neural network in granting credit.

In the second trial, the data was rearranged in different random order, and the first 100 applications were used to train the network. The remaining 25 applications were then evaluated using the trained network. The network misclassified 6 of the 25 applications in the sample for a 76% success rate. Classifications of good loans as bad, and of bad loans as good were equal at 12% each. The credit scoring method for this sample of 25 applications also misclassified 6 of the 25 applications [14].

INPUTS

Own/Rent your home.

Years with Employer.

Credit Cards.

Store Account.

Bank Account.

Occupation.

Previous Account.

Credit Bureau.

OUTPUT:

Credit Score.

3.2.2 Maximize Returns on Direct Mail with Neural Network

Microsoft, a leading computer software developer based in Redmond, Washington, is using BrainMaker neural network software to maximize returns on direct mail. Each year, Microsoft sends out about 40 million pieces of direct mail to 8.5 million registered customers. Most of these direct mailings are aimed at getting people to upgrade their software, or to buy other related products. Generally, the first mailing includes everyone in the database. The key is to send the second mailing to only those individuals who are most likely to respond [15].

Company spokesman Jim Minervino when asked how well BrainMaker neural network software had maximized their returns on direct mail responded, "Prior to using BrainMaker, an average mailing would get a response rate of 4.9%. By using BrainMaker, our response rate has increased to 8.2%. The result is a huge dollar difference that brings in the same amount of revenue for 35% less cost!"

To get a BrainMaker neural network to maximize returns on direct mail, several variables were fed into the network. The first objective was to see, which variables were significant and to eliminate those that were not. Some of the more significant variables were:

- Regency - the last time something was bought, and registered, calculated in number of days. It is a known fact that the more recently you've bought something, the better the chance you're going to buy more.
- First date to file - the date an individual made their first purchase. This is a measure of loyalty. The longer you've been a loyal customer, the better the chance is you're going to buy again.
- The number of products bought, and registered.
- The value of the products bought, and registered - figured at the standard reselling price.
- Number of days between the time the product came out, and when it was purchased, research has shown that people who tend to buy things as soon as they come out are the key individuals to be reached.

Additional variables include information taken from the registration card including yes/no answers to various questions - scored with either a one or zero - areas of interest like recreation, personal finances, and such personal information as age, and whether an individual is retired or has children. Microsoft also purchased data regarding the number of employees, place of employment, as well as sales and income data about that business. While Microsoft has designed this neural network for their own specific needs, some of these inputs could be applied to any network.

Prior to training, the information taken from the response cards was put into a format the network could use and yes/no responses were converted to numeric data. Minimums and maximums were also set on certain variables.

Initially, the network was trained with about 25 variables. To make sure the data was varied, it was taken from seven or eight campaigns and represented all aspects of the business including the Mac and Windows sides, from high and low price point products.

The model trained for about seven hours before it stopped making progress. At that point, variables that didn't have a major impact were eliminated. This process was repeated. Currently the model is based on nine inputs. Jim Minervino explains some of the other training considerations: "During training I used 'modify size' and I used 'prune neurons'; as training completes, I used 'add neuron', and we did an experiment with 'recurrent operations' although in the net model we ended up using the default [16]."

The output was a quantitative score from zero to one indicating whether an individual should receive or should not receive a second mailing. Minervino found that anybody scoring above 45 was more responsive to the mailing than anybody below.

The neural network was tested on data from twenty campaigns with known results not used during training. The results showed repeated and consistent savings. An average mailing resulted in a 35% cost savings.

3.2.3 Forecasting Required Highway Maintenance with Neural Networks

We've all driven on a road that is full of potholes or cracks. You can barely hold your commuter cup and you're anxious to get around that big semi so you can get into the smooth lane. But then you ask yourself; didn't they just fix this road last summer? Chances are you're right. But experienced highway maintenance engineers are hard to find, and as a result, the appropriate treatment isn't always selected.

Professor Awad Hanna at the University of Wisconsin in Madison has taken the guesswork out of the maintenance and repair process by training neural network to become a maintenance expert. If a seasoned professional isn't available, a recent college graduate and a computer program can do the job with a high degree of confidence. Since there is no mathematical formula to solve this kind of problem, it's an ideal application for neural network [16].

Professor Hanna trained the neural network with information provided by experts who can tell with a high degree of accuracy (confidence) which type of concrete is better than another for a particular problem. These experts were given a variety of situations and asked to provide various treatments. Professor Hanna then trained using the back propagation method on 1 hidden layer. Currently Professor Hanna is developing a simple program to be used with neural network that will take the input from the user and produce the most appropriate output based on previous experience provided by these senior people.

Some of the inputs include qualitative values for temperature and volume of a particular piece of pavement. Due to lack of funds, the number of input values was limited to 10. The output is the pavement treatment associated with a degree of confidence. For example, the recommended treatment might be chip seal with a confidence of 8 out of 10. Because there are so many variables, rarely is there a situation that occurs with 100% confidence [17].

While Professor Hanna's research is focused on a Midwestern area that experiences cold, ice and snow, and is based on the input of experts from this area only, his methodology could be applied to any geographic location. If human experts are not available to provide input, routine maintenance data from any Department of Transportation can be used instead. According to the Professor, Usually there is some kind.

3.2.4 A User Friendly Neural Network Trading System

Stock Prophet is a general purpose trading system development tool employing BrainMaker neural network technology to automatically combine multiple indicators into a single clear buy/sell signal. It can be applied to stocks, mutual funds, futures, and other financial instruments. Stock Prophet is a product of Future Wave Software.

1. Stock Prophet Highlights

Stock Prophet aids traders by consolidating multiple intermarket factors into a clear trading signal. Many market analysts have a repertoire of favorite indicators, but decision-making is difficult due to conflicting indications of market direction. Stock Prophet employs neural network technology to automatically combine multiple indicators into a single clear buy/sell signal. It does this by providing straightforward development of trading systems based on the artificial intelligence neural network technique as well as conventional technical analysis. The result is "institutional class" technical/quantitative analysis capability for the astute investor. Highlights of Stock Prophet are:

- Applicable to Stocks, Commodities, Mutual Funds, and Other Markets.
- Scientific Simulation Shows Extraordinary Profit Potential.
- Clear Signals Given Days, and Weeks **Before** Trade Execution Date. This is in contrast to essentially all technical indicators, which are late due to use of smoothing techniques.
- Complete Trading System can be Designed, Trained, and Tested for Profitability Within a Small Fraction of an Hour.
- Over 35 Indicators Plus Indicators of Other Indicators for an Explosive Number of Composite Indicators for Preprocessing.
- Helps to Select the Best Indicators by Analyzing Your Choice of Indicators for Ability to Predict Market Trend Using a Multiple Correlation Technique.
- Convenient Intermarket Capability Gives You the Edge.
- Automation MACRO Allows Easy Updates of Indicators.
- Provides Efficient Interface with BrainMaker, and Can Export to EXCEL. For IBM compatibles.

2. Stock Prophet's Forecasting System

As nearly all-neural network aficionados agree, the most difficult step in operating a neural network is gathering, and preprocessing voluminous, high-quality data. Neural networks, as powerful as they are, depend on applicable data in sufficient amounts, and in an appropriate format, to work their magic.

Stock Prophet automates much of the preprocessing needed to format data for BrainMaker while allowing the user to incorporate a wide array of well-known technical indicators.

In a 1995 issue of Technical Analysis of Stocks, and Commodities, Technical Editor John Sweeney noted that neural net integration is a major feature of Stock Prophet, saying that the user can "skip developing complex rules (and redeveloping them as their effectiveness fades) . . . just define the price series, and indicators you want to use, and the neural network does the rest [18]."

Moreover, as Sweeney goes on to say, "a major benefit of the neural network is that you don't have to define specific trading rules. Instead, the neural network derives the rules during training from the data. When BrainMaker sends an indicator back from its processing, the only rule required is above zero it's a buy, and below zero it's a sell."

Stock Prophet offers a full complement of technical indicators (on-balance volume, open interest, MACD, split volume, acceleration, etc) or the user can implement his or her own indicators by creating them in another program, and importing them via ASCII files. Additionally, Stock Prophet's data manipulation features include detrending, summation, limiting values, scaling, Fourier transformations, and biasing. Several of these indicators can be applied to previously generated indicators, thus increasing data preprocessing options. Many of these options (particularly the neural net data preprocessing features) can be automated through Stock Prophet's macro capability [18].

The value of a Stock Prophet/BrainMaker combination is summed up concisely by Stocks and Commodities' Sweeney: Stock Prophet's unique feature in developing trading signals is that it's neural net indicator is a prediction of future trend, published in advance of the trade date . . . if you get good signals 10 days in advance of the trade, you're going to be one delighted camper! If you could imagine that, try this program out.

3.3 Manufacturing Applications

3.3.1 Using Neural Networks to Determine Steam Quality

AECL Research in Manitoba, Canada has developed the INSIGHT steam quality monitor, an instrument used to measure steam quality, and mass flow rate. Steam Quality, and Mass Flow rate is the energy injected into the ground in an oil recovery project, for example. The improvement obtained by using the trained network was immediately apparent. Using a conventional linear program, the standard error of estimate (RMS of deviations about the ideal line) for steam quality, and mass flow rate are 28%, and 0.59 kg/s using the trained neural network, the standard error was 8.2%, and 0.34kg/s. A common test set of 26 sets of input data was used, and the network was trained on an additional 100 facts [19].

Later, a similar network was trained and tested all of the INSIGHT monitor calibration data obtained to date (i.e. data from tests at four different facilities collected over a period of seven years using a minimum of six to a maximum of nine different monitors). Here, the standard error of estimate for steam quality, and mass flow rate were 7.7%, and 0.4kg/s, respectively.

Recently AECL has successfully trained a neural network to return methanol, gasoline, and water contents from the RF reflectance spectra of mixtures of these three components. Currently they are investigating the application of a neural network to a-spectroscopy, and to the interpretation of on-line chemical sensor signals.

3.3.2 Neural Networks Optimize Enzyme Synthesis

A neural network has been trained to predict the outcome of a chemical reaction controlled by molar ratios, temperature, pressure, amount of enzyme, and stirring speed. Kirk, Barfoed, and Bjorkling at NOVO Nordisk A/S in Denmark used the BrainMaker program to train their neural network to predict the amount of desired product, and by-product, which would be formed after 22 hours of reaction time [20].

An excellent correlation between predicted yields, and experimental results was found. The neural network saves time, and money by predicting the results of chemical reactions so that the most promising conditions can then be verified in the lab, rather than performing a large number of experiments to gain the same information.

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

The neural network had three layers: 5 input layer neurons, 4 hidden layer neurons, and 2 output layer neurons. It was trained using the back propagation algorithm with the sigmoid threshold neuron function. Twelve facts were used to train the network to an accuracy of 96% for the outputs. In only a few minutes, all facts were learned. The trained network was then asked to make four predictions on data it hadn't seen before. The network predictions were compared to experimental observations. Very good correlations were found. The average deviation between the network, and the experiments was 4% (percentage of yield), ranging between 2%, and 7% difference. These deviations are within the normal experimental error of synthesis.

After being tested, the network was put to work evaluating thousands of possible conditions in order to find the most optimum. Using a simple algorithm, a test file was generated containing all of the possible values, totaling 9900 cases. The computer-generated test file contained values for each parameter which were both within, and without of the training value's range. The entire file ran through the network in 7 minutes, and the predictions were saved in a file. Using a search function, predictions for specified yields were selected. Only three cases were found to predict more than 88% monoester with a less than 4% formation of the digester. One of these cases was tested in the lab, and the results were close to experimental observation. The network had predicted 88.1% monoester, and the experiment yielded 86.2%. The network predicted 4.0% digester the experiment yielded 4.8%[20].

Finally, the 9900 predictions were again searched, but this time with additional restrictions more suitable for large-scale chemical processing. Again, the experimental results were very close to the yields predicted by the network.

3.3.3 Neural Network Optimizes IC Production by Identifying Faults

In many chip fabrication lines, an engineer analyzes failures to determine what could have caused failure. At Intel, this problem was previously attacked by an expert system, but this was found to be an inadequate tool, particularly in the case of multiple faults. Also, the expert system proved incapable of generalizing its knowledge, and completely hopeless with new cases.

Dan Seligson, Ph.D., used BrainMaker to create a neural network that could identify the fabrication problem that caused failures in finished Intel VLSI chips. The neural network was developed using information that was originally gathered for an expert system. The neural network was found to be 99.5% correct in generalizing data it had not seen before, but which was similar to that which it had. It was also found that the neural network was capable of distinguishing data, which was unlike any it had seen before (i.e. failures of 3 components in the system, when it had been trained with at most 2 failures) [21].

The original expert system was given the electrical test information from finished chips and the corresponding process control variables. The relationship between these two was determined by numerical experimentation and by simulation of CMOS process and device physics. A responsive surface model (RSM) was used to capture the results of exhaustive set of numerical experiments. Simulation tools were used to generate a database of paired sets of process variables and electrical test measurements. Rules were generated for the expert system from the e-test data and the corresponding process data [21].

BrainMaker was trained with the same pairs of e-test and process data. Eighteen e-test measurements were used as neural network inputs, and six process variables were the predicted outputs. A training error tolerance of 10% was used. E-test variables were categorized as one of 5 possible values from lowest to highest in order to determine classes of cases. Every legitimate e-test set must have its origin in a set of process variables so there are 15,625 (5^6) possible classed pairs of input/outputs. Of these, there are 24 classes of 1-fault cases and 240 classes of 2-fault cases. Actual continuous-valued numbers were used during training. 1500 examples of 2-fault cases were used, half for training, and half for testing the neural network. These cases were chosen randomly. An additional set of 100 3-fault cases was generated for testing the network's ability further. With an error tolerance of 20% (over the entire range of output values) the network correctly responded to 99.5% of the testing examples, indicating very good generalization. With an error tolerance of 10% fewer than 2% of the examples failed testing.

A lookup table was also implemented to solve this problem, but it was found that the network performance was superior. The problem was too large to include every single possible case (for either the expert system, the neural network, or the lookup table), so a sampling of data was used. A lookup table is unable to deal with non-linear changes in data as a neural network can.

3.3.4 Neural Network and Non-Destructive Concrete Strength Testing

In testing concrete for structural imperfections there are many different methods ranging from the drilling of core samples to the use of radar. The first method is destructive, time consuming, and allows for only a small percentage of the total area, while the second requires expensive equipment, and isn't effective when steel reinforcement is present. The National Institute of Standards, and Technology (NIST) has developed a non-destructive method for testing the internal structure of concrete [22].

Nondestructive testing (NDT) methods are used to obtain information about the properties or internal condition of an object without damaging it. Steel balls are dropped onto the concrete surface causing sound waves, which are reflected by cracks, and other imperfections in the concrete. These sound waves can then be collected, and analyzed by a neural network to determine the probability of a flaw. NIST has developed a system that used the thickness of the concrete as the base measurement, and was able to determine the depth of the flaw to 10% accuracy [22]. (The network was able to test a 0.4m thick slab with a 0.2m flaw, and determine that the flaw was 40% to 50% the depth of the slab

3.4 Medical Applications

3.4.1 Neural Network Reduces Expenses

A new hospital information and patient prediction system has improved the quality of care, reduced the death rate and saved millions of dollars in resources at Anderson Memorial Hospital in South Carolina. The CRTS/QURI system uses neural networks trained with BrainMaker to predict the severity of illness and use of hospital resources. Developed by Steven Epstein, Director of Systems Development and Data Research, the CRTS/QURI system's goal is to provide educational information and feedback to physicians and others to improve resource efficiency and patient care quality [23].

The first study showed that the program was directly responsible for saving half a million dollars in the first fifteen months even though the program only included half of the physicians and three diagnoses. Since then, the number of diagnoses, and physicians included in the program has increased. The quality of care has improved such that there are fewer deaths, fewer complications, and a lower readmission rate. Expenses have been reduced by fewer unnecessary tests and procedures, lowered length of stays, and procedural changes. The reported success has motivated several other hospitals to join in the program and has provided the impetus to begin a quality program with the state of South Carolina.

Individually trained neural networks learn how to classify and predict the severity of illness for particular diagnoses so that quality and cost issues can be addressed fairly. After attempts to use regression analysis to predict severity levels for several diagnoses failed, Epstein turned to the BrainMaker program for a new approach and taught his neural networks to classify and predict severity with 95% accuracy. The neural networks are also used to predict the mode of discharge - routine through death - for particular diagnoses.

Training information is based upon the length of stay in the hospital, which has a direct relationship to the severity of the illness (acuity). The neural network uses variables of seven major types: diagnosis, complications/comorbidity, body systems involved (e.g., cardiac and respiratory), procedure codes and their relationships (surgical or non-surgical), general health indicators (smoking, obesity, anemia, etc.), patient demographics (race, age, sex, etc.), and admission category. Three years of patient data was chosen for training. There were approximately 80,000 patients and 473 primary diagnoses. For a given diagnosis, about 400 to 1000 cases were used for training. Two neural networks for each diagnosis were trained - one to predict the use of resources and the other to predict the type of discharge. For a single diagnosis network, there are 26 input variables and one output variable [23].

3.4.2 Classify Breast Cancer Cells with Neural Network Software

A human who decides the degree of cancer present traditionally examines breast cancer cells under a microscope. People are inconsistent in these judgments from day to day, and from person to person.

A BrainMaker neural network that classifies breast cancer cells has been developed. The system was developed by Andrea Dawson MD of the University of Rochester Medical Center, Richard Austin MD of the University of California at San Francisco, David Weinberg, MD PhD of the Brigham, and Women's Hospital, and Harvard Medical School of Boston. Initial comparisons showed that BrainMaker is in good agreement with human observer cancer classifications [24].

Cancer cells are measured with the CAS-100 (Cell Analysis System, Elmhurst, IL). There are 17 inputs to the neural network, which represent morph metric features such as density, and texture.

There are four network outputs representing nuclear grading. The cancerous nucleus is graded as being well, moderate, or poorly differentiated, or as benign. Correct grade assignments were made between 52%, and 89% of the time on cases not seen during training. The lower success rate (for well differentiated) may have been due to the smaller percentage of this type in the training set. In addition, heterogeneity is much lower in well-differentiated tumors. Cancerous nuclei were classified within one grade of the correct grade.

3.4.3 Neural Network Orders Medical Laboratory Tests for Emergency Room

When a patient appears at the emergency room door it is sometimes an hour until a doctor can see him or her. It may be another hour until the lab can do the ordered tests. In order to save patient waiting time, Dr. Steven Berkov of Kaiser Hospital in Walnut Creek, California, developed a neural network program that can order the lab tests as soon as the patient is admitted. Up to 38 lab tests can be ordered by, the neural network [25].

Not only does the neural network save up to two hours of patient waiting time, it can reduce expenses. When the pilot system was tested, it reduced the number of tests that were ordered by 10-15%. Dr. Berkov says it could save half a million dollars a year. The neural network is able to reduce the number of tests for two reasons. First, medical records are used as examples for training the neural network. It can be determined which tests were actually necessary in retrospect, so the neural network can be trained to order only the pertinent tests. Second, nurses had been given the blanket permission to order tests and they tended to order even more than doctors.

The neural network has 67 inputs that include patient demographics and symptoms. This information is gathered when the patient is admitted and placed in the medical record. There are 38 outputs, each representing a different test that might be ordered. The pilot system neural network was trained on 250 patients from past hospital medical records.

When the pilot system was tested on new patients it was found to be about 95% accurate, according to Dr. Berkov. Most of the time the neural network would order most of the necessary tests. Sometimes it did not order enough, but Dr. Berkov explained that usually the doctor would only need to call the lab and order another test on the already collected specimen [25].

Inputs:

Age, sex, critical, serious, routine, ambulance, blood pressure, temperature, medications…etc.

Outputs:

Blood, gasses, acetone, CO_2, glucose, potassium, sodium liver… etc.

3.4.4 Diagnose Heart Attacks with BrainMaker Neural Network Software

When a patient complaining of chest pains is received by the emergency room, it is no simple matter to diagnose a heart attack. Merely examining the patient, and performing an electrocardiogram (EKG) is not often enough. If a patient is suspected of having experienced a heart attack, several blood samples are drawn over the next 4 to 48 hours. Patients with heart tissue damage will have various cardiac enzymes appear in their blood. There is a characteristic pattern of the change in enzyme levels during the short period after a heart attack. Using all three techniques (EKG, exam, and blood analysis), a doctor can diagnose, and treat heart attack patients. Neural network methods were found to correlate closely with expert human analysis, providing another opinion doctors can use to make a correct, and timely diagnosis.

A physician at St. Joseph Mercy Hospital in Michigan designed a neural network that recognizes cases of acute myocardial infarction (AMI, commonly called heart attack) using the cardiac enzyme data from series of tests on patients.[1] The input consisted of two sequential cardiac enzyme tests, and the number of hours between the tests. The output was "1" if the patient had a heart attack, and "0" if the patient did not. The network was trained with 185 examples from 47 patients using blood tests that were not more than 48 hours apart. There were a total of 21 inputs, and 1 output as shown below. The network was trained to a 10% error tolerance on all training data [26].

The neural network was then tested on 53 new sets of data. The data represented sets of serial cardiac enzyme data for ten patients with AMI, and eight patients without AMI. Neural network outputs of less than 10% probability of AMI were classified as no-AMI cases. Outputs of at least 90% probability of AMI were classified as AMI cases. Outputs between 10%, and 89% were interpreted as ambiguous or uncertain.

The neural network's diagnosis was then compared to three experts. One evaluated patients on the basis of ECHO/EKG changes. Another used the cardiac enzyme data. A third used autopsy reports. The network agreed with 100% of the AMI cases diagnosed by the cardiac enzyme expert, and 93% of the non-AMI cases. The 7% difference occurred where the network was uncertain. The network agreed with 86% of the AMI cases diagnosed by the EKG expert, and 33% of the non-AMI cases. In one case the EKG data was misleading due to multiple past heart attacks. In another case the network was uncertain. The network agreed with the autopsy results in 92% of the AMI cases, and 67% of the non-AMI cases. In one case the network was uncertain, and in another the heart had been damaged but by another cause.

3.4.5 Maximize Returns on Direct Mail with Neural Network Software

Microsoft, a leading computer software developer based in Redmond, Washington, is using BrainMaker neural network software to maximize returns on direct mail. Each year, Microsoft sends out about 40 million pieces of direct mail to 8.5 million registered customers. Most of these direct mailings are aimed at getting people to upgrade their software, or to buy other related products. Generally, the first mailing includes everyone in the database. The key is to send the second mailing to only those individuals who are most likely to respond.

Company spokesman Jim Minervino when asked how well BrainMaker neural network software had maximized their returns on direct mail responded, "Prior to using BrainMaker, an average mailing would get a response rate of 4.9%. By using BrainMaker, our response rate has increased to 8.2%. The result is a huge dollar difference that brings in the same amount of revenue for 35% less cost!"[27].

To get a BrainMaker neural network to maximize returns on direct mail, several variables were fed into the network. The first objective was to see which variables were significant, and to eliminate those that were not. Some of the more significant variables were:

- Regency - the last time something was bought, and registered, calculated in number of days. It is a known fact that the more recently you've bought something, the better the chance, you're going to buy more.
- First date to file - the date an individual made their first purchase. This is a measure of loyalty. The longer you've been a loyal customer, the better the chance is you're going to buy again.
- The number of products bought, and registered.
- The value of the products bought, and registered - figured at the standard reselling price.
- Number of days between the time the product came out, and when it was purchased. Research has shown that people who tend to buy things as soon as they come out are the key individuals to be reached.

Additional variables include information taken from the registration card including yes/no answers to various questions - scored with either a one, or zero - areas of interest like recreation, personal finances, and such personal information as age, and whether an individual is retired, or has children. Microsoft also purchased data regarding the number of employees, place of employment, as well as sales, and income data about that business. While Microsoft has designed this neural network for their own specific needs, some of these inputs could be applied to any network [27].

Prior to training, the information taken from the response cards was put into a format the network could use, and yes/no responses were converted to numeric data. Minimums, and maximums were also set on certain variables.

Initially, the network was trained with about 25 variables. To make sure the data was varied, it was taken from seven, or eight campaigns, and represented all aspects of the business including the Mac, and Windows sides, from high, and low price point products.

The model trained for about seven hours before it stopped making progress. At that point, variables that didn't have a major impact were eliminated. This process was repeated. Currently the model is based on nine inputs. Jim Minervino explains some of the other training considerations: 'During training I used 'modify size', and I used 'prune neurons'; as training completes, I used 'add neuron', and we did an experiment with 'recurrent operations' although in the net model we ended up using the default."

The output was a quantitative score from zero to one indicating whether an individual should receive, or should not receive a second mailing. Minervino found that anybody scoring above .45 was more responsive to the mailing than anybody below.

The neural network was tested on data from twenty campaigns with known results not used during training. The results showed repeated, and consistent savings. An average mailing resulted in a 35% cost savings.

3.4.6 Neural Network Predicts Functional Recovery

The Arcon Group provides accurate predictions of the functional recovery of patients over the Internet. These individual data based predictions are displayed in the form of line-graphs, and delivered to clinical personnel in a few seconds. The predictions lower hospital length-of-stay, improve sub acute, and home care outcomes, and significantly reduce the cost of patient care. They have proven invaluable for Quality Improvement, Resource Utilization, and managing care.

Arcon's FACT system predictions are derived from extensive research in the area of Rehabilitation Medicine, and the broad, and detailed medical database that resulted from it. The methodology incorporates state-of-the-art predictive power of Artificial Neural Nets, and global, instantaneous communication over the Internet [28].

In client hospitals where FACT is currently operating, length-of-stay has dropped thirty percent within the populous geriatric diagnosis related groups (DRG's) where functional recovery is a key determinant of hospital discharge.

The founder, and President, Loren M. Fishman M.D., Corporate Vice-President, Victor Oppenheimer Vice President of Legal, and Business Affairs, Marc L. Bailin, Esq., and other Arcon Group personnel develop purely data-driven tools valuable for improving institutional effectiveness, and efficiency, such as Arcon's FACT system.

Based in New York City, and Cambridge, Massachusetts, Arcon's advisory board includes medical, educational, business, and legal professionals of international distinction.

With its comprehensive mastery of emerging technologies, Arcon provides the health care industry with an accurate forecast of the course of functional recovery that is totally objective, and yet sensitive to each individual's uniqueness [28].

3.5 Science Applications

3.5.1 Neural Network Recognizes Mosquitoes in Flight

A neural network was trained to recognize two species, and both sexes of mosquitoes. The frequency of the wing beat is unique to each sex of each species. The neural network was given information about the wing beat frequency, and correctly classified the insects with a mean accuracy of 98%. Discriminate analysis had provided an accuracy rate of 84%. Even though the mosquitoes were of very similar species, the neural network had no trouble distinguishing them. Potential uses for this type of network include population/biological studies, pollination studies, evaluation of repellents, and attractant, *pest control, etc [28].*

Aubrey Moore of the Maui Agricultural Research Station, University of Hawaii, developed this network to assess the feasibility of automatically identifying insects in flight. A photo sensor was used to detect fluctuations in light intensity caused by reflections off individual mosquitoes flying through a light beam. Digital recording of the photo sensor signals was made with an analog-to-digital recorder. A change in light intensity triggered storage of 512 samples. Each signal was converted to a 256-wide frequency spectrum using a Fast Fourier Transform. One input was assigned for each of the 256 spectrum slices. One output was defined for each of the sex/species combinations for a total of four outputs [28].

The training set used 403 samples, approximately 100 for each sex/species combination. The network was tested on 57 samples, the species, and the network identified sex of every mosquito in the testing set correctly.

3.5.2 Neural Network Predicts Rainfall

The need for accurate local rainfall prediction is readily apparent when considering the many benefits such information would provide for river control, reservoir operations, forestry interests, flash flood watches, etc. While the data required to make such predictions has been available for quite some time, the complex, ever-changing relationships among the data, and its effect on the probability, much less the quantity, of rain has often proved difficult using conventional computer analysis. The use of a neural network, however, which learns rather than analyzes these complex relationships, has shown a great deal of promise in accomplishing the goal of predicting both the probability, and quantity of rain in a local area to an accuracy of 85%.

Using BrainMaker neural network software, Tony Hall (a hydro meteorologist from the National Weather Service in Fort Worth, Texas) has developed such a model. Nineteen meteorological variables (e.g. moisture, lift, instability, potential energy, etc.) were used to develop two networks for quantitative predictions--one for the warm season, and one for the cool season. Two additional networks for probability predictions were also generated. Another completely different program, written in C, was developed to allow both the quantitative, and the probability networks to run simultaneously with the results appearing on the same computer monitor [28].

Results to date have been outstanding. In the quantitative model, five categories were used to group the rainfall data (0.01 to 0.49 inches, 0.5 to 0.99 inches, 1.0 to 1.99 inches, etc.) Different tolerances were allowed for each range. For example, the tolerance for the first category was +/-0.2 inches while the tolerance for the higher categories ranged from 0.25 to 0.5 inches. Predictions for the quantitative models have been accurate in a range of 74% to 100% for the five categories with an overall accuracy of 83%.

The probability model used the criteria that a prediction of 30% probability or higher had to correspond to a rainfall of 0.10 inches or more. Otherwise the network output would be considered in error. The accuracy achieved to date for this model is 94% which, when combined with the quantitative results, gives an overall accuracy of 85%.

Sensitivity analysis was performed on the input variables to determine which had the most effect on the output. This will allow the developers to refine the models, and improve the accuracy. Since there are six additional sites in Texas that will be included in future studies, means of further automating both the data gathering, and BrainMaker operations are being investigated to improve the cost, and allow the technology to be used more economically.

3.5.3 Neural Network Predicts Detrimental Solar Effects

Dr. Henrik Lundstedt has trained neural networks to predict solar-terrestrial effects such as disturbances in the earth's magnetic fields. The disturbances have been known to cause blackouts, power plant shutdowns, corrosion in pipelines, disruptions in radio and television transmissions, malfunction of geological survey equipment, satellite tracking problems, and other detrimental effects. Being able to predict these occurrences helps prevent disasters [28].

The major cause of disturbances on earth are certain behaviors of the sun's solar wind, the solar wind is caused by several things such as coronal mass ejections or CMEs (which can trigger flares), and coronal holes. The neural network inputs consist of 37 known values of solar-terrestrial phenomena such as coronal mass ejections, coronal holes, solar sector boundaries, and proton events. The values are input as changes over the last four days. There are eight output neurons. The first output represents whether geomagnetic activity is expected to be quiet for the next day. The second, third and fourth outputs represent whether the activity is expected to be of a minor, major, or severe storm character. The fifth through eighth outputs predict the same items two days ahead [28].

3.5.4 Neural Network Processing for Spectroscopy

StellarNet Inc.'s moniker is "Intelligence from Light" -- an intriguingly cryptic way of describing the spectroscopic technology the Florida firm developed to optically analyze objects, and substances. StellarNet's Spectroscope bathes or permeates the sample being investigated with various lights, generating optical patterns called "spectra". Designed to identify the object or substance itself, and/or the presence, and concentrations of various components, Spectroscope uses a BrainMaker neural network in its SpectraNet application to process the spectral data, and make the appropriate recognition in real time [28]. Operating on PC hardware, and using a BrainMaker neural network as the processing engine, SpectraNet performs accurate, and detailed analysis in areas such as readout calibration for biomedical, environmental, and aerospace fiber optic monitoring sensors, chemical composition determination, quality assurance, process control, industrial monitoring, production control, and various trouble-shooting operations. In the agricultural area, one StellarNet customer is using SpectraNet's neural network capability to identify, and assure proper hydration in recently harvested onions.

SpectraNet automates BrainMaker training on inputs such as units of absorbance, transmittance, reflectance, chemical/biochemical composition, percent concentration, and relative irradiance, while incorporating full analysis capability for absorbance, transmittance, reflectance, or absorbance. While customizing neural net applications into turnkey instruments with various options for data acquisition, processing, and graphical display, SpectraNet will gather known spectral examples, analyze the data using quantitative measurements such as wet chemistry, and chromatography, and select the spectral regions for training (based on wavelength start, length, and increments). The data is then fed into the neural network for processing, and pattern recognition [28].

To help automate the data preparation process, StellarNet includes a SNAKE utility in all SpectraNet software packages that allows rapid spectral data configuration for training, and testing neural networks.

3.6 Sport Application

Selecting Winning Dogs with Neural Networks

Mr. Derek Anderson (Lakewood, CO) has trained neural networks that assist him in picking winning dogs at the racetrack. He trained the neural networks with two months of race results found in the daily racing booklets. Once trained, he runs the current day's race information through seven neural networks. He adds up the dogs' "scores" from his neural networks, and places them in predicted finish order. Whenever the first place dog is ahead by at least ten neural network points over the second place dog, he bets on the winner. He claims 94% accuracy with this method, but he can bet on only a third of the races [28].

Mr. Anderson input information for approximately 300 races for the training file. The neural network looks at the statistics for three dogs at a time and outputs which of the three dogs did best. If there are eight dogs in a race, he must group the dogs in all possible combinations of three: dogs A, B, and C; dogs A, B, and D; dogs A, B, E; etc. For each race, there are 56 combinations, or sets of input data.

The data Mr. Anderson uses include the winning time of the race, the time that each dog took to finish the race, the time that dog reached each of four positions in the race (out of box, first corner, backstretch, outside corner) as well as comments about the dog's behavior. The behavior was classified as one of fifteen types such as ran wide, bumped, hit, and ran inside. He presented these pieces of information for each dog for each of the last eight races the dog ran. His networks have 504 inputs (21 statistics * 3 dogs * 8 races = 504).

Mr. Anderson has designed six basic neural networks with these 504 inputs. The difference in the six networks is the output. One neural network has three outputs which represent which dog is best: dog A, dog B, or dog C. The dog, which was best, gets 1 the others get a 0. Another has three outputs, which represent which dog did worst of the three. The dog, which did the worst, gets a 0, the others get a 1. Another network has three outputs, which represent whether the dog was in the top three finishing positions [28].

These dogs get a 1; the others get a 0. An opposite network outputs if the dog was not in the top three. Another pair of networks output whether the dog was in the last three to finish the race.

Because the dog racing information is not available in computer format, Derek spent a lot of time doing data entry. When it's time to predict a race, Derek runs the data through all of his networks, and adds up the score for each dog. The scores range from 0 to 25 most of the time. The dog with the highest score is the winner.

INPUTS:

1st race, dog A: win time.

1st race, dog A: finish time.

1st race, dog A: 1st corner time.

1st race, dog A: backstretch time.

1st race, dog A: outside corner.

1st race, dog A: behavior.

1st race, dog B: win time.

(etc for 8 races, dogs A B, and C).

OUTPUTS:

Dog A rating, dog B rating, dog C rating.

3.7 Summary

This chapter presents several applications of neural networks. We have considered the business applications, where a neural network gives the best facility to manage very complex problems in business field. We have discussed the applications in science where we have put the points on the alphabetic to judge now, that a neural network is the most important factor, which affect the variety of the science improvement. We have discussed the application of a neural network in manufacturing, we conclude how much important to apply neural network to see more, and more fussy, and cheap products. Finally we can notice that there is no limit for the neural networks applications.

CHAPTER FOUR

Neural Network for Pattern Recognition

4.1 Overview

Different types of neural network models are applicable when dealing with pattern recognition concept. The scope, and purpose of neural networks are becoming increasingly popular in pattern recognition.

Many applications are achieved via neural network models, but in this chapter the interesting thing is considering the pattern recognition applications in details.

Pattern recognition concept was the hardest problem according to the researchers and engineers, because of the lack of traditional computers' abilities to solve such complicated problems, however, the neural computers offers the facilities to deal with such problems and solving them easily.

Almost any neural network applications would fit into pattern recognition area. There is some potential for neural network recognition purposes, including special resource allocation, analysis, and scheduling.

So it is not possible to report all of pattern recognition neural applications. But let's start taking a view to some of million of these applications.

4.2 Decoding Algorithms and Predicting Sequences

The ability to predict data sequences is important in data transmission to provide error correction. Certain algorithms can predict repetitive code with good accuracy, but fail in the presence of noisy code sequences.

Mr. James Johnson of Netrologic, Inc. (Dayton, OH) trained a neural network on noisy data and was able to predict code sequence accuracy from 62% to 93%, depending upon the initial conditions and the presence or absence of noise. Higher accuracy could probably be obtained by training a network with a wider variety of training samples. The network was given an input of 100 bits generated using this algorithm:

b (a) = b (a-3) XOR b (a-31) where 32^{2} a ^{2} 100.

The network was asked to predict what the 101st bit should be in that sequence with no explicit knowledge of how the string was formed. The equation used to generate the bits contained a 31-bit random seed. A set of 1,000 training facts was generated to train a back propagation net. The first data sets were generated with sets of correlated data; that is, five sets of 100 bits were generated using the algorithm above, and a 31-bit seed that was identical except that it was shifted right one additional position for each subsequent set of data to generate five separate sets of 100 bits. Then a new random 31-bit seed was generated, and five more correlated 100-bit sets were produced.

The network learned all of the 1,000 training sets to within 10%. A test set was generated of 500 sets of 100 strings. The network got 468 out of 500 correct [29].

4.3 Chaos, Strange Attractors and BrainMaker Plots

Take the last 200 years' data on cotton production. Plot a point, which is one years' production versus the next years'. You get data points scattered all over the screen like stars at night. If you were to plot a lot of points (without lines connecting them) you get a shape, like a donut. The points seem to fall on or near a circle. This is a strange attractor.

In a Normal or Real attractor, you get dense collection of points in the middle and spreading out fading out. The price has equilibrium; the production has equilibrium, represented by the dense collection around a single point. A Strange attractor is an attractor for which there is not an equilibrium point [30].

There is no math currently that explains the plot of something versus something else which produces the donut. The presence of a Strange attractor means you're dealing with a chaotic system. A chaotic system is a nonlinear feedback system. In the chaotic cotton production system, what you learn by seeing the Strange attractor is that there is some sort of a feedback mechanism, there is an analytic solution to what the system is doing and there is feedback around the analytic solution.

You get Strange attractors when you look at the population of foxes over the years as it grows and shrinks. This is chaotic, rather than random. In a random system, you get points scattered all over with no shape whatsoever and there is no underlying mechanism, therefore no way to predict anything. In a chaotic system there is an underlying mechanism with no linearity and feedback. It is believed by some that, because there is an underlying mechanism analytic approaches can be used to make predictions.

With BrainMaker Professional you can make plots to find Strange attractors. In Net Maker you put cotton price in a column, cotton price shifted down by one in another, plot one on the X and one on the Y. Plot lots of months worth of data. You will see a donut, a Strange attractor, which indicates an underlying mechanism with non-linearity and feedback. If you discover the underlying math that explains this, please call us immediately.

4.4 Neural Networks Recognize Chemical Drawings

Pattern recognition is a commonly encountered problem when computers are required to get information from the physical world around them. It may be easy enough to get a digital picture via a camera or a scanner into a computer file, but how does the computer know what the data means? Recent advances in commercially available optical character recognition software have provided some affordable solutions, particularly when fonts are similar and the material is relatively clean. Blind people can even purchase a scanner and software, which will read aloud to them. However, there are still real limits to what most commercial software can recognize. Most have difficulty when the print is sloppy, small or varies considerably. None offer the ability to recognize arbitrary shapes, symbols or graphics.

Recent studies in pattern recognition with neural networks have been sponsored by the US Post Office to read ZIP codes [31]. Even though they are primarily interested in hand-written digits, the techniques developed are general. Feature extraction from bitmaps is the biggest problem. An approach for feature extraction uses Fourier descriptors of the items to be recognized. One such application, described here, reads a chemical drawing (comprised of characters and graphics) and translates it into a chemical structure database.

Compounds are described in two ways: as a chemical drawing of connected atoms, or as a list of atoms and their connections in a connection table. A connection table can be easily stored on computer, but most printed sources such as books, journals and papers use the more easily recognized drawings. The connection tables uniquely define compounds and can be used to index information in a database. When chemical compound descriptions are placed in a database with other information they can be used for patent searches, environmental studies, toxicology studies, and precursor searching, for example.

Fein-Marquart Associates, Inc. has developed a program, which automatically reads printed chemical drawings and translates them into connection tables in a database. The old approach required manual computation of the connection table. Commercially available optical character recognition programs were not able to read the chemical drawings because many use a very small print (6 and 8 point) and there are graphic elements present as well as Standard English characters.

The system was developed by Fein-Marguart and uses a neural network trained with BrainMaker Professional to recognize the printed characters and graphics. The system has a 98% recognition success rate according to Joe McDaniel, Senior Staff Member at Fein-Marquart. The chemical drawings are read into a PC from a scanner, some mathematical processing is performed to provide Fourier descriptors, which are then fed into a neural network for recognition and translation into bonds and atomic symbols. The output of the neural network is formatted into a connection table and transmitted to a host computer database.

Fourier descriptors are computed by tracing the outline of a character to create a concave hull. This data is stored as a list of x and y coordinates. If one views the x portion of the data as the real and the y as the imaginary portion of a complex data pair, and then performs a Fourier transform on the list, the result will be a list of complex data points representing frequency. Straight lines or big curves can be interpreted from low frequency data, and corners, serif and end-of-lines from high frequency data. Characters and graphics have frequency magnitude and phase signatures, which can be recognized by the neural network.

Low frequency data can be interpreted as straight lines or big curves, and high frequency data as corners, serif and end-of-lines. Characters and graphics have frequency magnitude and phase "signatures" which can be recognized by the neural network. The neural network is given the frequency information as input and is trained to translate information into bonds and atomic symbols.

The output of the neural network is formatted into a connection table and transmitted to a host computer database. When chemical compound descriptions are placed in a database with other information, they can be used for patent searches, environmental studies, toxicology studies, and precursor searching [32].

4.5 Neural Networks Provide Context for OCR

Neural networks offer a general-purpose solution to pattern recognition problems. They are able to generalize much better than traditional programs and can run faster. Neural networks are not limited to any set of characters, and can learn to recognize just about anything, even things like tools, mechanical parts, aircraft, and cancerous cells.

Neural networks are also useful in determining context in conjunction with traditional OCR applications. For example, when reading a book or journal a neural network can look at the words and tell you if it's reading a title, an author, a publisher, or a date. It has been difficult to get traditional programs to quickly provide such contextual information.

Electronic Data Publishing, Inc. (Brooklyn, NY) has incorporated a neural network into its OCR/database system. The system reads documents such as journals and papers, and places information into a database for later retrieval into reports or catalogs. The neural network classifies the material read in from an OCR program into categories such as author, title, abstract, publisher or date, so that it can be tagged and stored in a database for later retrieval. "The neural network has saved $20,000 of labor costs in the first two months and allows the same number of people to get four times as much data through the system," said Ken Blackstein, designer of the neural network. The printed material contains too many variations in the data to be effectively classified using a Prolog decision tree. The neural network approach was chosen for its ability to generalize well when given ample data. [33]

This neural network is one of the largest, most successful designs known. The 1440-input, 20-output network was trained with 200 megabytes of data using neural network running on the neural network accelerator board. After roughly 100 training runs, the neural network converged to 96% accuracy on all training examples. In the three months of use with new data, the neural network has made no errors.

Prior to being read by a scanner, the material is photocopied, perhaps enlarged, and cleaned up by people who may also use a felt pen to block out extraneous printed material. The printed pages are then scanned into a PC with the OmniPage (Caere Corporation) OCR program under the Windows environment. The overall system is depicted in figure 1. The words are then processed through the Soundex algorithm, which reduces the number of characters and produces a "word" which is similar to a phoneme. This helps the neural network to generalize, because nearly identical printed words such as "Johnson," "Jonson," and "Johnsen," will appear the same to the neural network. This also reduces the number of inputs to the neural network because Soundex "words" are comprised of fewer characters than English words. The design is similar to Sejnowski's famous "NetTalk", except that a full line of text is input rather than seven characters, and the output is a classification rather than a phoneme for speech production.

The output of the neural network is used to place the text into database developed with Netware (Novell, Inc.). Currently, medical literature is on-line with 600 megabytes of data, which is roughly equivalent to 200,000 pages of printed information. Electronic Data Publishing, Inc. has plans for an Engineering database, which would require the training of another neural network that understands engineering terms.

4.6 Neural Network Recognizes Voice Mail

By now, everyone is familiar with voice mail technology. You call a business and a voice directs you to use your touch-tone phone to direct your call or to leave a voice message. Of course if you don't have a touch-tone phone, the current voice mail technology isn't accessible, and you need to wait for the operator to help you - unless the system contains a neural network.

By using BrainMaker to train neural networks for speech recognition, Dr. Mark Ortner of Compass Technology in Sarasota, Florida has developed P.C. based software that will make voice recognition technology affordable to small and medium sized businesses around the world. Soon, you'll be able to reach the voice mailbox of the desired party by phone or fax whether you have touch-tone phone or not.

In 1992, Compass Technology was acquired, by Octel Communications the world's largest provider of voice and fax information processing and services. Currently, Dr. Mark Ortner is revolutionizing information processing. So far, Dr. Ortner has trained a network using 2500 facts and 28 words, including the numbers "zero" through "nine", the words, "yes" and "no", and the names of various departments within the company. The degree of recognition accuracy ranges from 90-97%.

The current application is for voice recognition of an extension. Dr. Ortner collects "voice data" by having a variety of callers dial an extension and enter a "mailbox" located at the "voice training center." The callers' words are run through a normalizer (confidential in nature), which creates a BrainMaker output file (fact file). That information is then trained on a neural net.

According to Dr. Ortner, the advantages of this kind of system are far-reaching. In the United States, only about 37% of the telephones are not touch-tone. However in most other countries, touch-tone is nonexistent. By providing a trainable voice recognition system, the purchaser could record the words used at their location, train the network, then run it. A caller could say, "Extension 230" and the system would make the correct phone transfer. Touch-tone is no longer necessary.

Dr. Ortner's network has 400 input neurons, 107 hidden neurons and, at the present time, the output layer has 28 neurons. This will eventually change as more words are added. The output of the network is the recognition of the spoken word, which is drawn from a symbol table.

Dr. Ortner uses BrainMaker for training the neural net, but wrote the software that actually runs the neural net. (BrainMaker Professional comes with this code as part of the Runtime License). Since the largest system Compass Technology handles is 32 ports, this kind of voice recognition system is ideal for small and medium sized businesses.

Once this system is on the market, Dr. Ortner will turn his attention to developing a phonic-based system. The caller will be able to pronounce a word and have the system convert it directly to text. This would be a big advantage to the deaf. A deaf person could read a voice message as it is printed on screen or print a message back and have it translated into voice [33].

4.7 Summary

In this chapter we have described many applications of pattern recognition neural application, the most important thing is we should now understand that neural networks give us the ability to face any problem resulted by the pattern recognition concept and can be solved by the models of neural networks without wasting time and money.

We have discussed how neural networks now can decode the algorithm and predict the sequences; we have seen the Chaos and Strange attractor and neural plots. The interesting thing we have explained was how the neural nets can recognize the chemical drawing, and now we can summarize our chapter by taking in mind that neural network is only the best solver for any problem you face in the technology fields.

CONCLUSION

The neural network contains a large number of simple neuronlike processing elements and a large number of weighted connections between the elements. The weights on connections encode the knowledge of a network. Though biologically inspired, many of neural network models developed do not duplicate the operation of human brain. The intelligence of a neural network emerges from the collective behavior of neurons, each of which performs only limited operation. Neural networks can only learn if the training set consists of good examples, the learning ability is determined by the neural network's architecture and by the algorithmic method chosen for training. Neural networks represent a new technology with many potential uses. Their capabilities have already been proven in a variety of, manufacturing, business, science, sport, medicine, and pattern recognition.

First chapter explained the background and the history of the neural networks, which shows the developments and improvements through previous years. Also the advantages that made the neural network new technology in order to be used in many successful applications.

We can conclude a neural network is a concept of processing data based on the way neurons in the brain process information. Neural network can organize, classify convert, and learn pattern. Neural networks are being used in: investment analysis, signature analysis, process control, monitoring, marketing, forecasting… etc.

In chapter two the architectures of neural networks were described in details such as perceptron, Hopfield network, back-propagation, and Hopfield. A perceptron model can be trained and make decision. During the training phase, pairs of input and output vectors are used to train the network. Back-propagation is a multi-layer network can be trained using back-propagation learning algorithm. This involves the presentation of pairs of input and output vectors [5].

A Hopfield network is used with binary inputs, which are initialized using training samples (input and output). In the decision making phase, the test data is presented to the network at a certain time [5].

Also the way of learning of neural networks were explained, supervised learning and unsupervised learning. In supervised learning process, the input data and its output are presented to the neural network, and according to a defined law, the neural network changes its weights in order to reproduce the correct output vector when an input vector is applied. Unsupervised-learning process requires only input vectors to train the network. Once the input data is presented to the neural network, the weights are adjusted in ordered way according to some defined figure of merit [5].

The use of neural networks offers many properties and capabilities, such as nonlinearity, input output mapping, adaptivity, evidential response and fault tolerance. For example a neural network, implemented in hardware form, has the potential to be inherently fault tolerant in these in the sense that its performance is degraded gracefully under adverse operating conditions.

Chapter three was aimed to show the fields, where the neural networks can be applicable. So many applications of neural network either in real world or predicted applications also used in many specific paradigms: business, science, medicine, and sports. One of these applications was the most interesting applications, that is neural network predicts detrimental solar effects, and many applications cannot be limited.

Chapter four was the final chapter; it is aimed to determine the applications of neural network specifically in pattern recognition. Some applications were described such as decoding algorithm and predicting sequence, recognizing chemical drawing, providing contents for OCR, and recognizing voice mail.

REFERENCES

[1] Alexander Bain, “Mind and Body The Theories of Their Relation” 1873-Alexande.

[2] Radolf Kolliker, “Manual of Human Histology” 1853-Radulf.

[3] 1963 Paper R.o. of Winder, http://www.calsci.com/intro ToNN.html

[4] Jelinek, Herbert F., and Eduardo Femandez. 1998. Neurons and fractals: how reliable and useful are calculations of fractal dimensions? Journal of Neuroscience Methods, 81:9-18.

[5] Adnan Khashman, Lecture Notes of COM420, “Neural Network”, N.E.U.

[6] J.Anderson, “A simple Neural Generating an Interactive Memory” 14:197-220,1972.

[7] Aihara, K., T. Takabe, M. Toyda. 1990. Chaotic Neural Networks. Physics Letters A144: 333-340.

[8] Schi\oler, H. and Hartmann, U. (1992) "Mapping Neural Network Derived from the Parzen Window Estimator", Neural Networks, 5, 903-909.

[9] An Introduction to Neural Computing. Alexandria, and Morton, H. 2nd edition, http://www.calsci.com/intro ToNN.html

[10] “Introduction to Neural Networks by Jeannette Lawrence”, http://www.calsci.com/intro ToNN.html

[11] Generation5: Artificial Intelligence Repository-Introduction-, http://library.thinkquest.org/1824/nnintro.shtml

[12] Neural Networks at pacific Northwest National Laboratory, http://www.emsl.pnl.gov:2080/docs/cie/neural/neural.homepage.html

[13] Eberhart RC and Dobbins RW (1990) Early Neural Network Development History: The Age of Camelto. IEEE Engineering in Medicine and Biology 9, 15-18.

[14] Jagielska I. (1993), "The application of neural networks to business information systems", Proceedings of 4th Australian Conference on Information Systems, Brisbane, pp. 565-574.

[15] Windsor C. G. & Harker A. H. (1990), "Multi-variate financial index prediction - a neural network study", Proceedings of International Neural Network Conference, Paris, France, pp. 357-360.

[16] Wang H. A. & Chan A. K.-H. (1993), "A feedforward neural network model for Hang Seng Index", Proceedings of 4th Australian Conference on Information Systems, Brisbane, pp. 575-585.

[17] Ling C. S. (1993), "Choosing the right neural network model for trading", Proceedings of First Symposium on Intelligent Systems Applications, Singapore, pp. 26-33.

[18] Eberhart R. C. & Dobbins R. W. (1990), Neural Network PC Tools - A Practical Guide, San Diego: Academic Press.

[19] C.H. Messom, C.J. Hinde, A.A. West, D.J. Williams. 1992. "Designing Neural Networks for Manufacturing Process-Control Systems." Proceedings of the 1992 IEEE International Symposium on Intelligent Control, Glasgow, Scotland.

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

[21] "Intelligent Manufacturing Trough Neural Networks - Monitoring of Machining Processes." Intelligent Autonomous Systems IAS-3, Vol.? Ch. 72, pp. 328-337.

[22] Makridakis et al. (1983), Forecasting: "Methods and Applications", New York: Wiley.

[23] Neural Networks at Pacific Northwest National Laboratory, http://www.emsl.pnl.gov:2080/docs/cie/neural/neural.homepage.html

[24] Artificial Neural Networks in Medicine, http://www.emsl.pnl.gov: 2080/docs/cie/techbrief/NN.techbrief.ht

[25] DARPA Neural Network Study (October, 1987-February, 1989). MIT Lincoln Lab. Neural Networks, Eric Davalo and Patrick Naim.

[26] MeDavid Brown. 1994. "Artificial Neural Networks for Biomedical Applications." The Workshop of Computer-aided. Diagnosis in Medical Imaging, Georgetown University Medical Center, Washington, DC, (5-6 July 1994).

[27] H. Burke. 1994. "A Computerized Prediction System for Cancer Survival that Uses an Artificial Neural Network." First World Congress on Computational Medicine, Public Health and BioTechnology, Austin, TX, 25-28 April.

[28] Data and Analysis Center for Software "Artificial Neural Network Technology", 1992 http:// www.docs.dtic.mil/techs/neural/neural.litle.html

[29] PatterG. J. McLachlan, Discriminant Analysis and Statistical Pattern Recognition, New York: John Wiley, 1992. (Not for the beginner: oriented towards a statistical audience: many results on parametric models, especially Gaussian case: encyclopedic collection of references to other work, many obscure references to statistical papers on pattern recognition which can not be found in other texts).

[30] K.Fukunaga, Introduction to Statistical Pattern Recognition, 2nd Ed., New York: Academic Press, 1990. (Thorough review of similar material to that covered in Duda and Hart but including many results derived between 1973 and 1990: does not include discussion of more recent classification methods such as neural networks or decision trees).

[31] P. A. Devijver and J. Kittler, Pattern Recognition: A Statistical Approach, Englewood Cliffs, NJ: Prentice-Hall International, 1980. (Covers similar material to Fukunaga and Duda and Hart, well written for the beginner).

[32] J. Hertz, A. Krogh, and R. G. Palmer, Introduction to the Theory of Neural Computation, Redwood City, CA: Addison-Wesley, 1991. (Well-written text on neural networks, available in paperback: however, written from a physicist's viewpoint and thus does not emphasize neural network connections with statistical pattern recognition very strongly).

[33] D. J. Hand, Discrimination and Classification, Chichester, UK: John Wiley and Sons, 1981. (Very well-written text emphasizes statistical aspects of pattern recognition but is well suited to beginners, many examples, written at an intuitive level).

YAKIN DOĞU ÜNİVERSİTESİ
FEN-EDEBİYAT FAKÜLTESİ
TÜRK DİLİ VE EDEBİYATI BÖLÜMÜ

MEZUNİYET ÇALIŞMASI:

ZİRAAT MECMUASI
LEFKOŞA DEVLET MATBAASI NİSAN 1928
(94.-107. SAYFALAR ARASI)

İSMAİL YILDIRIM
20012055

DANIŞMAN HOCA: YRD.DOÇ.DR.ALİ EFDAL ÖZKUL

LEFKOŞA 2006

İÇİNDEKİLER

ÖNSÖZ

Hazırlamış olduğum bu çalışma Lefkoşa Devlet Matbaası tarafından 1928 yılının Nisan ayında tefrika edilmiştir.Girne Milli Arşivinde muhafaza edilen bu eseri Yakın Doğu Üniversitesi, Fen-Edebiyat Fakültesi, Türk Dili ve Edebiyatı Bölümü, 2005-2006 eğitim ve öğretim yılına ait mezuniyet çalışması olarak 94. sayfadan 107. sayfaya kadar olan bölümün çevirisini ben üstlendim.

Yapmış olduğum bu çalışmada K.K.T.C.'nin ziraat ve tarım ürünleri, bu ürünlerin yetiştirilmesi, bu ürünlerin hastalıkları ve bu hastalıkların tedavisi, bu ürünlere zarar veren böceklerin yaşamları ve korunma yöntemleri, bu ürünlerin ticareti (alım-satım) gibi konular ele alınmıştır.Bu eser detaylı bir şekilde incelendiğinde K.K.T.C.'nin tarım ve ziraatı hakkında geniş bilgi sahibi olunacaktır.

Bu çalışmamda bana yol gösteren sayın hocam **Yard.Doç.Dr.Ali Efdal Özkul**'a sonsuz teşekürlerimi sunarım.

İsmail YILDIRIM