NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

APPLICATIONS OF NEURAL NETWORKS

Graduation Project
COM- 400

Student: Ahmed Qatanani

Supervisor: Assoc.Prof. Dr Adnan Khashman

Nicosia - 2001

ACKNOWLEDGEMENT

First of all I would like to thank Allah Almighty who gave me the strength to complete this project.

Then I would like to thank my Supervisor Assoc.Prof.Dr. Adnan Khashman who is not only a teacher but also a very appreciating personality.

Although I face many difficulties finding appropriate data regarding some topics of the project, but he (Dr.Adnan) has guided me with keen interest and helped me by all means. Dr.Adnan, thanks for your continual support.

I would like to thank all our teachers in the Near East University, Student Advisors and all the Staff of the Faculty of Engineering for giving us the opportunity to experience this remarkable Institute that have showed us preliminary steps towards our professional carrier.

Many thanks to my parents for their continuous moral support and encouragement.

Also many thanks to my friends especially Ahmed El taha, Ahmed wadi, Mohammed Abu shaban and both of my cousins Ata and Ashraf.

ABSTRACT

Technology of Neural Network gives a computer system an amazing capacity to actually learn from input data. Artificial neural networks have provided solutions to problems normally requiring human observation and thought processes. Neural network simulations appear to be a recent development. However, this field was established before the advent of computers, and has survived at least one major setback and several eras. The computing world has a lot to gain fron neural networks. Their ability to learn by example makes them very flexible and powerful.

The most basic components of neural networks are modeled after the structure of the brain. Some neural network structures are not closely to the brain and some does not have a biological counterpart in the brain. However, neural networks have a strong similarity to the biological brain and therefore a great deal of the terminology is borrowed from neuroscience. Given this description of neural networks and how they work, what real world applications are they suited for? Neural networks have broad applicability to real world business problems. In fact, they have already been successfully applied in many industries, science, medicine, Manufacturing, and Sports...etc.

Neural networks are trained by repeatedly presenting examples to the network. Each example includes both inputs and outputs. The network tries to learn each of your examples in turn, calculating its output based on the inputs you provided. Neural networks represent a new technology with many potential uses. Their capabilities have already been proven in a variety of business applications. Almost any neural network application would fit into one business area or financial analysis. There is some potential for using neural networks for business purposes, including resource allocation and scheduling.

INTRODUCTION

Chapter one aimed to show the background of the neural networks and what the reasons and the benefits of using neural networks. And also how the neural networks improved to be new technology in the present and the future. Neural network is defined as massively parallel-distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects: (1) Knowledge is acquired by the network through a learning process, and (2) Interneuron connection strengths known as synaptic weights are used to store the knowledge.

Neural network simulations appear to be a recent development. However, this field was established before the advent of computers, and has survived at least one major setback and several eras. Many important advances have been boosted by the use of inexpensive computer emulations. Neural networks learn by example. They cannot be programmed to perform a specific task.

The development of true Neural Networks is a fairly recent event, which has been met with success. The future of Neural Networks is wide open, and may lead to many answers and/or questions.

Chapter two describes the architectures and basic components of neural networks; the most basic components of neural networks are modeled after the structure of the brain. Some neural network structures are not closely to the brain and some does not have a biological counterpart in the brain. However, neural networks have a strong similarity to the biological brain and therefore a great deal of the terminology is borrowed from neuroscience. Neural networks are named after the cells in the human brain that perform intelligent operations. The brain is made up of billions of neuron cells. Each of these cells is like a tiny computer with extremely limited capabilities; however, connected together, these cells form the most intelligent system known. Neural networks are formed from hundreds or thousands of simulated neurons connected together in much the same way as the brain's neurons.

The term `architecture' has been much abused in the history of mankind. It has many meanings depending on whether you are talking about buildings, inside of computers or neural networks among other things. Even in neural networks, the term architecture and what we have been referring to as `type' of neural network are used interchangeably. So when we refer to such and such architecture, it means the set of possible interconnections (also called as topology of the network) and the learning algorithm defined for it.

Also Learning algorithms, which were considered for a single perceptron, linear adaline, and multilayer perceptron belong to the class of supervised learning algorithms. Two basic groups of unsupervised learning algorithms and related self-organizing neural networks, namely: Hebbian Learning, Competitive Learning.

Networks such as the one just described are called artificial neural networks (ANNs), in the sense that they represent simplified models of natural nerve or neural networks.

Chapter three shows the fields where the Neural Networks can be applied, Neural Networks are performing successfully where other methods do not, recognizing and matching complicated, vague, or incomplete patterns

Neural networks have been applied in solving a wide variety of problems. The most common use for neural networks is to project what will most likely happen. There are many applications of Neural Networks that can be applied in the real world. Although one may apply neural network systems for prediction, diagnosis, planning, monitoring, debugging, repair, instruction, and control, the most successful applications of neural networks are in categorization and pattern recognition.

A number of real applications can also be found in the NeuroForecaster package. Based on these successful applications, it is therefore evident that the neural network technology can be applied to many real-world problems especially those related to business, financial and engineering modeling.

Chapter four highlights the important application of Neural Network in business, management and finance, the application would fit into one business area or financial analysis. There is some potential for using neural networks for business purposes, including resource allocation and scheduling. There is also a strong potential for using neural networks for database mining, that is, searching for patterns implicit within the explicitly stored information in databases. Most of the funded work in this area is classified as proprietary. Thus, it is not possible to report on the full extent of the work going on. Most work is applying neural networks, such as the Hopfield-Tank network for optimization and scheduling.

CHAPTER ONE
BACKGROUND
ON
NEURAL NETWORKS

1.1 Overview

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

It would be the most remarkable thing ever. And all humans can live happily ever after. This is the aim of artificial intelligence in general. Neural network simulations appear to be a recent development. However, this field was established before the advent of computers, and has survived at least one major setback and several eras. Many important advances have been boosted by the use of inexpensive computer emulations. The future of Neural Networks is wide open, and may lead to many answers and/or questions. Is it possible to create a conscious machine? What rights do these computers have? How does the human mind work? What does it mean to be human?

Neural networks process information in a similar way the human brain does. The network is composed of a large number of highly interconnected processing elements (neurons) working in parallel to solve a specific problem. Neural networks learn by example. They cannot be programmed to perform a specific task. The examples must be selected carefully otherwise useful time is wasted or even worse the network might be functioning incorrectly.

1.2 Background of Neural Networks

Many tasks, which seem simple for us, such as reading a handwritten note or recognizing 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 analyzes data by passing it through several simulated processors which are interconnected with synaptic-like "weights". Although the programming and mathematics behind neural network technologies are complex, using neural network software can be quite simple and the results are often quite extraordinary. Once you have collected several records of the data you wish to analyze, the network will run through them and "learn" how the inputs of each record may be related to the result. Each "record" might be a machine on an assembly line, or a particular stock, or the weather one day.

If the record was a patient at a hospital, the record's inputs (such as: age, sex, body fat, allergies, blood pressure) and it's related output (such as: did the drug work in this case?) are both fed into the "neurons" of the network. The network then continually refines itself until it can produce an accurate response when given those particular inputs. After training on a few dozen cases, the network begins to organize 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 internal mapping of weights that can accurately reproduce the expected output. Once you realize 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. [4]

1.3 What is a Neural Network?

There are definitions of Neural Network

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

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

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

Neural networks: is a buzz word. 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 modeling - and are strongly supported by industry.

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

Neural Network: technology gives a computer system an amazing capacity to actually learn from input data. Artificial neural networks have provided solutions to problems normally requiring human observation and thought processes.

Neural network: is powerful computational tools that can be used for classification, pattern recognition, empirical modeling and for many other tasks. Neural networks (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 modeled.

Neural Network: is a system loosely modeled on the human brain. The field goes by many names, such as connectionism; parallel distributed processing, neuron-computing, natural intelligent systems, machine learning algorithms, and artificial neural networks. It is an attempt to simulate within specialized hardware or sophisticated software, the multiple layers of simple processing elements called neurons.

Each neuron is linked to certain of its neighbors with varying coefficients of connectivity that represent the strengths of these connections. Learning is accomplished by adjusting these strengths to cause the overall network to output appropriate results. [13]

1.4 Historical background of Neural Network

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

These pioneers were able to develop convincing technology, which surpassed the limitations identified by Minsky and Papert. Minsky and Papert, published a book (in 1969) in which they summed up a general feeling of frustration (against neural networks) among researchers, and was thus accepted by most without further analysis. Currently, the neural network field enjoys a resurgence of interest and a corresponding increase in funding.

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

1. **First Attempts:** There were some initial simulations using formal logic. McCulloch and Pitts (1943) developed models of neural networks based on their understanding of neurology. These models made several assumptions about how neurons worked. Their networks were based on simple neurons, which were considered to be binary devices with fixed thresholds. The results of their model were simple logic functions such as "a or b" and "a and b". Another attempt was by using computer simulations. Two groups (Farley and Clark, 1954; Rochester, Holland, Haibit and Duda, 1956). The first group (IBM researchers) maintained closed contact with neuroscientists at McGill University. So whenever their models did not work, they consulted the neuroscientists. This interaction established a multidisciplinary trend, which continues to the present day.
2. **Promising & Emerging Technology:** Not only was neuroscience influential in the development of neural networks, but psychologists and engineers also contributed to the progress of neural network simulations. Rosenblatt (1958) stirred considerable interest and activity in the field when he designed and developed the *Perceptron*. The Perceptron had three layers with the middle layer known as the association layer. This system could learn to connect or associate a given input to a random output unit. Another system was the ADALINE (*ADAptive Linear Element*), which was developed in 1960 by Widrow and Hoff (of Stanford University). The ADALINE was an analogue electronic device made from simple components. The method used for learning was different to that of the Perceptron, it employed the Least-Mean-Squares (LMS) learning rule.

3. **Period of Frustration & Disrepute:** In 1969 Minsky and Papert wrote a book in which they generalized the limitations of single layer Perceptrons to multilayered systems. In the book they said: "...our intuitive judgment that the extension (to multilayer systems) is sterile". The significant result of their book was to eliminate funding for research with neural network simulations. The conclusions supported the disenchantment of researchers in the field. As a result, considerable prejudice against this field was activated.
4. **Innovation:** Although public interest and available funding were minimal, several researchers continued working to develop neuromorphically based computational methods for problems such as pattern recognition. During this period several paradigms were generated which modern work continues to enhance. Grossberg's (Steve Grossberg and Gail Carpenter in 1988) influence founded a school of thought, which explores resonating algorithms. They developed the ART (Adaptive Resonance Theory) networks based on biologically plausible models. Anderson and Kohonen developed associative techniques independent of each other.

Klopf (A. Henry Klopf) in 1972, developed a basis for learning in artificial neurons based on a biological principle for neuronal learning called *heterostasis*. Werbos (Paul Werbos 1974) developed and used the *back-propagation* learning method, however several years passed before this approach was popularized. Back-propagation nets are probably the most well known and widely applied of the neural networks today.

In essence, the back-propagation net. Is a Perceptron with multiple layers, a different threshold function in the artificial neuron, and a more robust and capable learning rule? Mari (A. Shun-Ichi 1967) was involved with theoretical developments: he published a paper, which established a mathematical theory for a learning basis (error-correction method) dealing with adaptive pattern classification.

While Fukushima (F. Kunihiko) developed a stepwise trained multilayered neural network for interpretation of handwritten characters. The original network was published in 1975 and was called the *Cognitron.*

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

Today: Significant progress has been made in the field of neural networks-enough to attract a great deal of attention and fund further research.

Advancement beyond current commercial applications appears to be possible, and research is advancing the field on many fronts. Neurally based chips are emerging and applications to complex problems developing. Clearly, today is a period of transition for neural network technology. [4]

1.5 Why use neural networks?

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

This expert can then be used to provide projections given new situations of interest and answer "what if" questions.
Other advantages include:

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

2. Self-Organization: An ANN can create its own organization or representation of the information it receives during learning time.
3. Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
4. Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

1.6 Neural networks versus conventional computers

Neural networks take a different approach to problem solving than that of conventional computers. Conventional computers use an algorithmic approach i.e. the computer follows a set of instructions in order to solve a problem.

Unless the specific steps that the computer needs to follow are known the computer cannot solve the problem. That restricts the problem solving capability of conventional computers to problems that we already understand and know how to solve.

But computers would be so much more useful if they could do things that we don't exactly know how to do.

Neural networks process information in a similar way the human brain does. The network is composed of a large number of highly interconnected processing elements (neurons) working in parallel to solve a specific problem.

Neural networks learn by example. They cannot be programmed to perform a specific task. The examples must be selected carefully otherwise useful time is wasted or even worse the network might be functioning incorrectly.

The disadvantage is that because the network finds out how to solve the problem by itself, its operation can be unpredictable.

On the other hand, conventional computers use a cognitive approach to problem solving; the way the problem is to solve must be known and stated in small unambiguous instructions.

These instructions are then converted to a high-level language program and then into machine code that the computer can understand.

These machines are totally predictable; if anything goes wrong is due to a software or hardware fault. Neural networks and conventional algorithmic computers are not in competition but complement each other. There are tasks are more suited to an algorithmic approach like arithmetic operations and tasks that are more suited to neural networks.

Even more, a large number of tasks require systems that use a combination of the two approaches (normally a conventional computer is used to supervise the neural network) in order to perform at maximum efficiency.

1.7 What are Neural Networks Used For?

Their applications are almost limitless but fall into a few simple categories.

Classification: Customer/Market profiles, medical diagnosis, signature verification, loan risk evaluation, voice recognition, image recognition, spectra identification, property valuation, classification of cell types, microbes, materials, samples.

Forecasting: Future sales, production requirements, market performance, economic indicators, energy requirements, medical outcomes, chemical reaction products, weather, crop forecasts, environmental risk, horse races, jury panels.

Modeling: Process control, systems control, chemical structures, dynamic systems, signal compression, plastics molding, welding control, robot control, and many more. [13]

1.8 Who Needs Neural Network?

People that have to work with or analyze data of any kind. People in business, finance, industry, education and science whose problems are complex, laborious, fuzzy or simply un-resolvable using present methods. People who want better solutions and wish to gain a competitive edge.

- Computer scientists want to find out about the properties of non-symbolic information processing with neural nets and about learning systems in general.
- Statisticians use neural nets as flexible, nonlinear regression and classification models.
- Engineers of many kinds exploit the capabilities of neural networks in many areas, such as signal processing and automatic control.
- Cognitive scientists view neural networks as a possible apparatus to describe models of thinking and consciousness (High-level brain function).
- Neuro-physiologists use neural networks to describe and explore medium-level brain function (e.g. memory, sensory system, motorics).
- Physicists use neural networks to model phenomena in statistical mechanics and for a lot of other tasks.
- Biologists use Neural Networks to interpret nucleotide sequences.

Philosophers and some other people may also be interested in Neural Networks for various reasons.

1.9 Past and Present

The development of true Neural Networks is a fairly recent event, which has been met with success. Two of the different systems (among the many) that have been developed are: the basic feedforward Network and the Hopfield Net. Neural networks are 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 modeling - and are strongly supported by industry. There is a lot of up-to-date information available on the WWW and we have listed important web sites, which will open the doors for you to a very exciting field of research. [13]

1.10 Future of Neural Network

The future of Neural Networks is wide open, and may lead to many answers and/or questions. Is it possible to create a conscious machine? What rights do these computers have? How does the human mind work? What does it mean to be human? Because gazing into the future is somewhat like gazing into a crystal ball, so it is better to quote some "predictions". Each prediction rests on some sort of evidence or established trend which, with extrapolation, clearly takes us into a new realm.

Prediction1: Neural Networks will fascinate user-specific systems for education, information processing, and entertainment. "Alternative realities", produced by comprehensive environments, are attractive in terms of their potential for systems control, education, and entertainment. This is not just a far-out research trend, but is something, which is becoming an increasing part of our daily existence, as witnessed by the growing interest in comprehensive "entertainment centers" in each home. This "programming" would require feedback from the user in order to be effective but simple and "passive" sensors (e.g. fingertip sensors, gloves, or wristbands to sense pulse, blood pressure, skin ionization, and so on), could provide effective feedback into a neural control system. This could be achieved, for example, with sensors that would detect pulse, blood pressure, skin ionization, and other variables, which the system could learn to correlate with a person's response state.

Prediction2: Neural networks, integrated with other artificial intelligence technologies, methods for direct culture of nervous tissue, and other exotic technologies such as genetic engineering, will allow us to develop radical and exotic life-forms whether man, machine, or hybrid.

Prediction3: Neural networks will allow us to explore new realms of human capability realms previously available only with extensive training and personal discipline. So a specific state of consciously induced neurophysiologically observable awareness is necessary in order to facilitate a man machine system interface. [32]

1.11 Are there any limits to Neural Networks?

The major issues of concern today are the scalability problem, testing, verification, and integration of neural network systems into the modern environment. Neural network programs sometimes become unstable when applied to larger problems.

The defense, nuclear and space industries are concerned about the issue of testing and verification. The mathematical theories used to guarantee the performance of an applied neural network are still under development. The solution for the time being may be to train and test these intelligent systems much as we do for humans.

Also there is some more practical problems like: The operational problem encountered when attempting to simulate the parallelism of neural networks. Since the majority of neural networks are simulated on sequential machines, giving rise to a very rapid increase in processing time requirements as size of the problem expands. Solution: implement neural networks directly in hardware, but these need a lot of development still. Instability to explain any results that they obtain. Networks function as "black boxes" whose rules of operation are completely unknown.

1.12 Advantages of Neural Network

An advantage is that the programmer doesn't need to feed the system with expert knowledge about the model. All the network needs to have is some input data along with the preferred output.

1. They deal with the non-linearities in the world in which we live.
2. They handle noisy or missing data.
3. They create their own relationship amongst information - no equations!
4. They can work with large numbers of variables or parameters.
5. They provide general solutions with good predictive accuracy.

1.13 Disadvantages of Neural Network

Most neural networks don't work with probabilities. This implies that the answer given by a system working on a neural network cannot be connected with a given probability. Nor can the system calculate what the second best answer is. For a number of applications this is adequate, but for a decision support system, the probabilities of the answer should be provided. Furthermore alternative answers should be available from the system. Another problem with neural networks is that it is impossible to follow the reasoning behind a given answer.

In a decision support system it is desirable to know the arguments for an answer, so the operator can verify that the choice has been made on sound reasoning from the system. A neural network has to go through a training phase before it can be taken into use. If the programmer cant get lots of examples of the behavior he requires from the system, the system cant be trained properly.

CHAPTER TWO
ARCHITECTURES
OF
NEURAL NETWORKS

2.1 Overview

Neural networks should be applied in situations where traditional techniques have failed to give satisfactory results, or where a small improvement in modeling performance can make a significant difference in operational efficiency or in bottom-line profits. Direct marketing is an excellent example of where a small improvement can lead to significant results. The response rate on direct marketing campaigns is usually quite low. A five percent response rate is often considered very good.

The human nervous system, it is now known, consists of an extremely large number of nerve cells, or neurons, which operate in parallel to process various types of information. Neurocomputing involves processing information by means of changing the states of networks formed by interconnecting extremely large numbers of simple processing elements, which interact with one another by exchanging signals. Networks such as the one just described are called artificial neural networks (ANNs), in the sense that they represent simplified models of natural nerve or neural networks.

An individual wanting to investigate this emerging technology and explore ways in which it can improve his/her organization is advised to consult with neural network practitioners who have experience in developing and implementing models for use in commercial applications. The bottom line is that any manager interested in getting more useful information from available data should consider neural network technology as an option. They can be used by aggressive organizations to focus available resources more effectively, thus gaining a valuable competitive edge.

2.2 Neural Network Computing

The majority of information processing today is carried out by digital computers. This has led to the widely held misperception that information processing is dependent on digital computers.

However, if we look at cybernetics and the other disciplines that form the basis of information science, we see that information processing originates with living creatures in their struggle to survive in their environments, and that the information being processed by computers today accounts for only a small part - the automated portion - of this.

Viewed in this light, we can begin to consider the possibility of information processing devices that differ from conventional computers. In fact, research aimed at realizing a variety of different types of information processing devices is already being carried out, albeit in the shadows of the major successes achieved in the realm of digital computers. One direction that this research is taking is toward the development of an information-processing device that mimics the structures and operating principles found in the information processing systems possessed by humans and other living creatures.

Digital computers developed rapidly in and after the late 1940's, and after originally being applied to the field of mathematical computations, have found expanded applications in a variety of areas, to include text (word), symbol, image and voice processing, i.e. pattern information processing, robot control and artificial intelligence. However, the fundamental structure of digital computers is based on the principle of sequential (serial) processing, which has little if anything in common with the human nervous system. [16]

The human nervous system, it is now known, consists of an extremely large number of nerve cells, or neurons, which operate in parallel to process various types of information. By taking a hint from the structure of the human nervous system, we should be able to build a new type of advanced parallel information processing device.

In addition to the increasingly large volumes of data that we must process as a result of recent developments in sensor technology and the progress of information technology, there is also a growing requirement to simultaneously gather and process huge amounts of data from multiple sensors and other sources.

This situation is creating a need in various fields to switch from conventional computers that process information sequentially; to parallel computers equipped with multiple processing elements aligned to operate in parallel to process information.

Besides the social requirements just cited, a number of other factors have been at work during the 1980's to prompt research on new forms of information processing devices. For instance, recent neuropsychological experiments have shed considerable light on the structure of the brain, and even in fields such as cognitive science, which study human information processing processes at the macro level, we are beginning to see proposals for models that call for multiple processing elements aligned to operate in parallel.

Research in the fields of mathematical science and physics is also concentrating more on the mathematical analysis of systems comprising multiple elements that interact in complex ways. These factors gave birth to a major research trend aimed at clarifying the structures and operating principles inherent in the information processing systems of human beings and other animals, and constructing an information processing device based on these structures and operating principles. The term "neurocomputing" is the name used to refer to the information engineering aspects of this research. [15]

2.3 The Biological Foundation of NeuroComputing

Neurocomputing involves processing information by means of changing the states of networks formed by interconnecting extremely large numbers of simple processing elements, which interact with one another by exchanging signals. Networks such as the one just described are called artificial neural networks (ANNs), in the sense that they represent simplified models of natural nerve or neural networks.

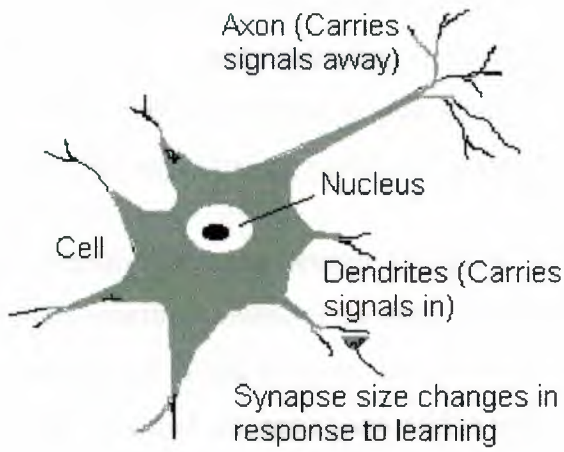


Figure 2.1. A simple neuron cell

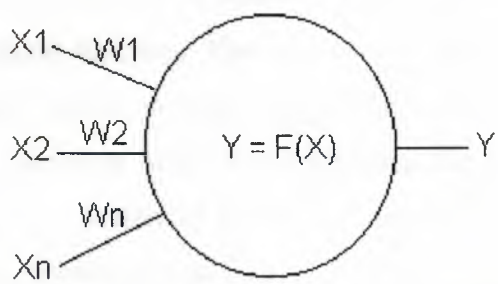


Figure 2.2. A schematic diagram of a neuron

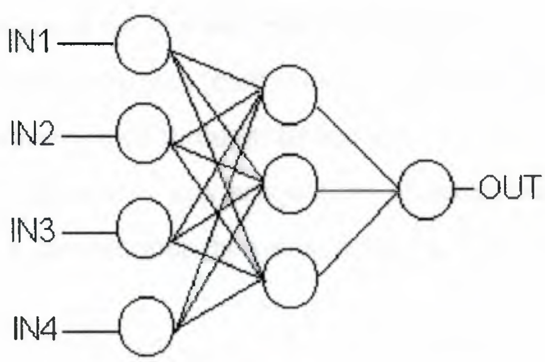


Figure 2.3. A feed forward neural network

The basic processing element in the nervous system is the neuron. The human brain is composed of about 1011 of over 100 types.

Tree-like networks of nerve fiber called dendrites are connected to the cell body or soma, where the cell nucleus is located. Extending from the cell body is a single long fiber called the axon, which eventually branches into strands and sub strands, and are connected to other neurons through synaptic junctions, or synapses.

The transmission of signals from one neuron to another at synapses is a complex chemical process in which specific transmitter substances are released from the sending end of the junction. The effect is to rise to lower the electrical potential inside the body of the receiving cell. If the potential reaches a threshold, a pulse is sent down the axon - we then say the cell has "fired".

In a simplified mathematical model of the neuron, the effects of the synapses are represented by "weights" which modulates the effect of the associated input signals, and the nonlinear characteristics exhibited by neurons is represented by a transfer function which is usually the sigmoid function. The neuron impulse is then computed as the weighted sum of the input signals, transformed by the transfer function. The learning capability of an artificial neuron is achieved by adjusting the weights in accordance to the chosen learning algorithm, usually by a small amount *Wj = **Xj where * is called the learning rate and * the momentum rate. [13]

2.4 Brain plasticity

- At the early stage of the human brain development (the first two years from birth) about 1 million synapses (hard-wired connections) are formed per second.
- Synapses are then modified through the learning process (plasticity of a neuron).
- In an adult brain the above may account for plasticity two mechanisms: creation of new synaptic connections between neurons, and modification of existing synapses.

2.5 What can you do with an NN and what not?

In principle, NNs can compute any computable function, i.e., they can do everything a normal digital computer can do. In practice, NNs are especially useful for classification and function approximation/mapping problems which are tolerant of some imprecision, which have lots of training data available, but to which hard and fast rules (such as those that might be used in an expert system) cannot easily be applied.

Almost any mapping between vector spaces can be approximated to arbitrary precision by feed forward NNs (which are the type most often used in practical applications) if you have enough data and enough computing resources.

To be somewhat more precise, feed forward networks with a single hidden layer, under certain practically-satisfied assumptions are statistically consistent estimators of, among others, arbitrary measurable, square-integral regression functions, binary classifications.

NNs are, at least today, difficult to apply successfully to problems that concern manipulation of symbols and memory. And there are no methods for training NNs that can magically create information that is not contained in the training data.

2.6 Taxonomy of neural networks

From the point of view of their active or decoding phase, artificial neural networks can be classified into feed forward (static) and feedback (dynamic, recurrent) systems.

From the point of view of their learning or encoding phase, artificial neural networks can be classified into supervised and unsupervised systems.

2.6.1 Feed forward supervised networks

These networks are typically used for function approximation tasks. Specific examples include:

- Linear recursive least-mean-square (LMS) networks
- Back propagation networks
- Radial Basis networks.

2.6.2 Feed forward unsupervised networks

These networks are used to extract important properties of the input data and to map input data into a ``representation'' domain. Two basic groups of methods belong to this category.

- Hebbian networks performing the Principal Component Analysis of the input data, also known as the Karhunen-Loeve Transform.
- Competitive networks used to performed Learning Vector Quantization, or tessellation of the input data set. Self-Organizing Kohonen Feature Maps also belong to this group. [26]

2.7 The Analogy to the Brain

The most basic components of neural networks are modeled after the structure of the brain. Some neural network structures are not closely to the brain and some does not have a biological counterpart in the brain. However, neural networks have a strong similarity to the biological brain and therefore a great deal of the terminology is borrowed from neuroscience.

2.7.1 The Biological Neuron

The most basic element of the human brain is a specific type of cell, which provides us with the abilities to remember, think, and apply previous experiences to our every action. These cells are known as neurons; each of these neurons can connect with up to 200000 other neurons. The power of the brain comes from the numbers of these basic components and the multiple connections between them. All natural neurons have four basic components, which are dendrites, soma, axon, and synapses. Basically, a biological neuron receives inputs from other sources, combines them in some way, performs a generally nonlinear operation on the result, and then output the final result. The figure below shows a simplified biological neuron and the relationship of its four components.

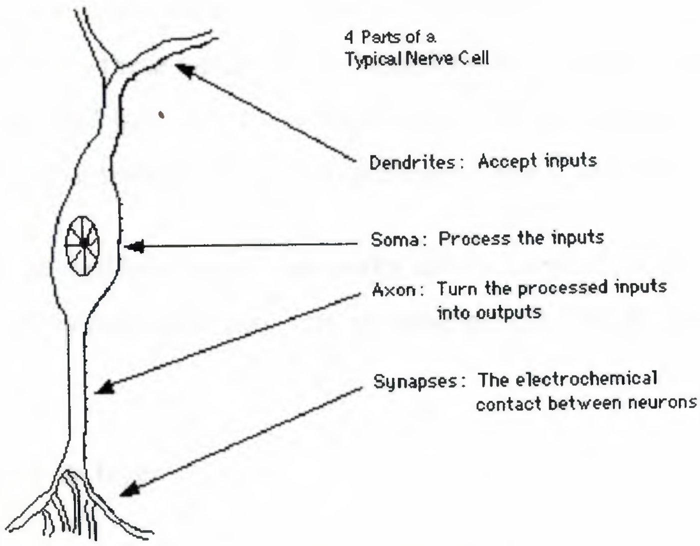


Figure 2.4. A simplified biological neuron

2.7.2 The Artificial Neuron

The basic unit of neural networks, the artificial neurons, simulates the four basic functions of natural neurons. Artificial neurons are much simpler than the biological neuron; the figure below shows the basics of an artificial neuron.

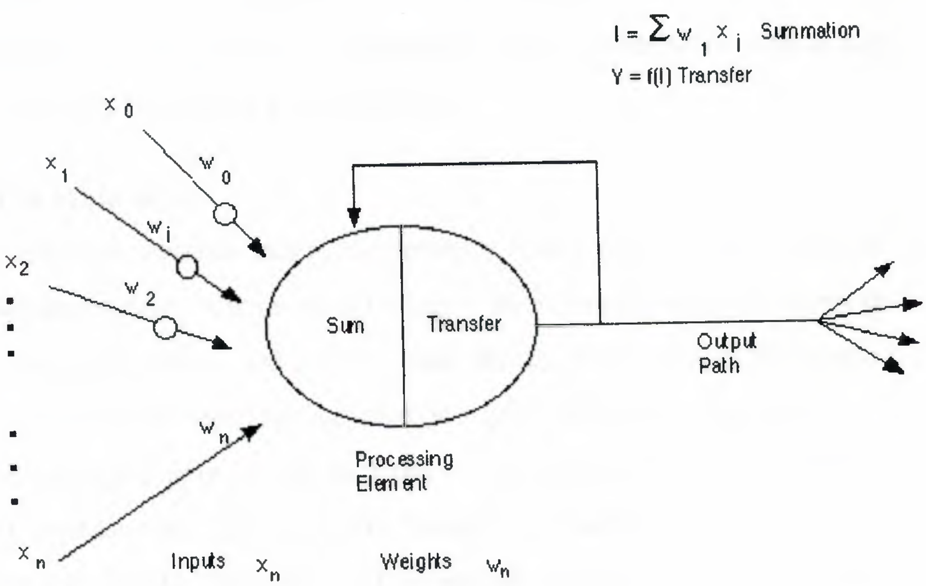


Figure 2.5. The basics of an artificial neuron

Note that various inputs to the network are represented by the mathematical symbol, x (n). Each of these inputs are multiplied by a connection weight, these weights are represented by w (n). In the simplest case, these products are simply summed, fed through a transfer function to generate a result, and then output.

Even though all artificial neural networks are constructed from this basic building block the fundamentals may vary in these building blocks and there are differences.

2.7.3 Neural layer on layer

Neural systems consist of layers of neurons that are connected to each other. Typically, there are three layers: an input layer, an output layer, and a hidden layer. One type of neural system architecture that I have used for financial forecasting is known as a feed forward network with supervised learning.

This type of system has two or more layers, with neurons in one layer receiving information only from the previous layer and sending outputs only to the next layer. Neurons in a given layer do not interconnect. Each neuron in a layer is connected to every neuron of the succeeding layer, with mathematical weights (or connection strengths) assigned to their connections. This is known as "fully connected" network configurations.

2.7.3.1 The input layer

The input layer presents data to the network. The number of data categories determines the number of neurons in the input layer. Each category of input data requires one input neuron, and it is here that the size and structure of the neural system must be determined. For instance, in a S&P 500 or DJIA prediction system, if your input data include each day's closing price for the Deutschemark, S&P 500, Japanese yen, Treasury bills, Eurodollars, Swiss franc, U.S. dollar index, Treasury bonds, DJIA and gold, as well as the discount and Fed funds rates (a total of 12 categories of data), your network would have 12 neurons in the input layer. Massaging the data with moving averages, ratios and so on to eliminate data noise will affect the number of input neurons.

Coupled with each day's input data would be the next day's S&P 500 or DJIA closing price. Each of these input/output pairs of data or training pattern is called "fact."

The hidden layer is composed of neurons that are connected to neurons in the input and output layers but do not connect directly with the outside world. The hidden layer is where the system recodes the input data into a form that captures the hidden correlations, allowing the system to generalize from previously learned facts to new inputs.

Experimentation often determines the number of hidden layers and the appropriate number of neurons in them. Too few neurons impair the network and prevent it from correctly mapping inputs to outputs, while too many neurons impede generalization by allowing the network to "memorize" the patterns presented to it without extracting any of the salient features (similar to curve-fitting or over optimization).

Then, when presented with new patterns, the network cannot process them properly because it has not discovered the underlying relationships.

2.7.3.2 The output layer

Each neuron in the output layer receives its inputs from each neuron in the hidden layer. Your desired output determines how many output neurons the system needs.

Each output category requires one output neuron. Thus, if we want to predict the next day's open, high, low and close for the S&P 500 or the DJIA, the system would, in fact, need four neurons in the output layer. With supervised learning, you would provide the neural system with "facts" that represent input training patterns (today's prices, discount rate and Fed funds rate) that you expect the system to encounter subsequently during trading, and an output-training pattern (next day's prices) that you want it to forecast.

In this manner, during training the system forecasts as its output the next day's S&P 500 or DJIA level, which is then used to adjust each neuron's connection weight, so that during subsequent training iterations, the system will be more likely to forecast the correct output.

For the system to learn during training there must be a way to alter the connection weights in terms of how much and in which direction they will be changed. This algorithm, or paradigm, is known as the "learning law." While numerous learning laws can be applied to neural systems, perhaps the most widely used is the generalized delta rule or back propagation method.

During each iteration of training, the inputs presented to the network generate "a forward flow of activation" from the input to the output layer. Then, whenever the output forecast by the system (next day's S&P 500 or DJIA) is incorrect when compared with its corresponding value in the training pattern, information will flow backward from the output layer to the input layer, adjusting the weights on the inputs along the way. On the next training iteration, when the system is presented with the same input data, it will be more likely to forecast the correct output.

The learning law for a given network defines precisely how to modify these connection weights between neurons to minimize output errors during subsequent training iterations. If no error occurs, then no learning is needed for that fact. Eventually, when the system has completed learning on all of the facts, it reaches a stable state and is ready for further testing.

2.8 Design

The developer must go through a period of trial and error in the design decisions before coming up with a satisfactory design. The design issues in neural networks are complex and are the major concerns of system developers.

Designing a neural network consist of:

- Arranging neurons in various layers.
- Deciding the type of connections among neurons for different layers, as well as among the neurons within a layer.
- Deciding the way a neuron receives input and produces output.
- Determining the strength of connection within the network by allowing the network learns the appropriate values of connection weights by using a training data set.

2.8.1 Layers

Biologically, neural networks are constructed in a three dimensional way from microscopic components. These neurons seem capable of nearly unrestricted interconnections. This is not true in any man-made network. Artificial neural networks are the simple clustering of the primitive artificial neurons. This clustering occurs by creating layers, which are then connected to one another. How these layers connect may also vary. Basically, all artificial neural networks have a similar structure of topology. Some of the neurons interface the real world to receive its inputs and other neurons provide the real world with the network's outputs. All the rest of the neurons are hidden form view.

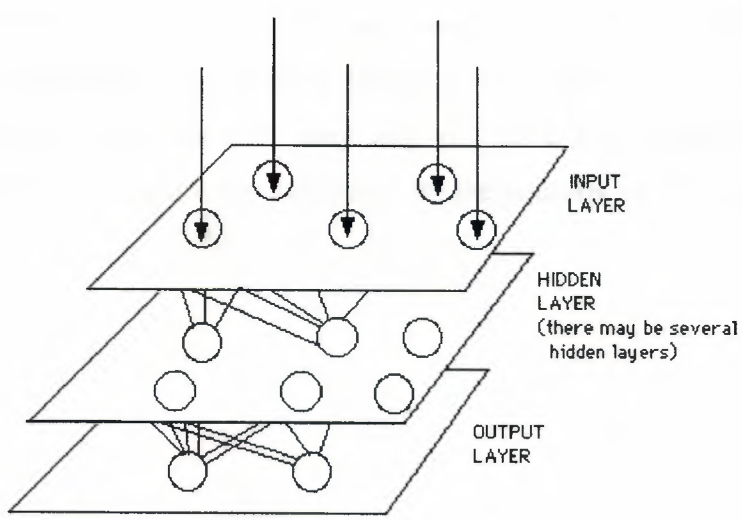


Figure 2.6. Artificial neural network

As the figure above shows, the neurons are grouped into layers. The input layer consists of neurons that receive input form the external environment.

The output layer consists of neurons that communicate the output of the system to the user or external environment. There are usually a number of hidden layers between these two layers; the figure above shows a simple structure with only one hidden layer. When the input layer receives the input its neurons produce output, which becomes input to the other layers of the system. The process continues until a certain condition is satisfied or until the output layer is invoked and fires their output to the external environment. To determine the number of hidden neurons the network should have to perform its best, one are often left out to the method trial and error. If you increase the hidden number of neurons too much you will get an over fit, that is the net will have problem to generalize. The training set of data will be memorized, making the network useless on new data sets.

2.8.2 Communication and types of connections

Neurons are connected via a network of paths carrying the output of one neuron as input to another neuron. These paths is normally unidirectional, there might however be a two-way connection between two neurons, because there may be a path in reverse direction.

A neuron receives input from many neurons, but produce a single output, which is communicated to other neurons. The neuron in a layer may communicate with each other, or they may not have any connections. The neurons of one layer are always connected to the neurons of at least another layer.

2.8.2.1 Inter-layer connections

There are different types of connections used between layers; these connections between layers are called inter-layer connections.

- **Fully connected**

  Each neuron on the first layer is connected to every neuron on the second layer.
- **Partially connected**

  A neuron of the first layer does not have to be connected to all neurons on the second layer.
- **Feed forward**

  The neurons on the first layer send their output to the neurons on the second layer, but they do not receive any input back form the neurons on the second layer.
- **Bi-directional**

  There is another set of connections carrying the output of the neurons of the second layer into the neurons of the first layer.

Feed forward and bi-directional connections could be fully- or partially connected.

- **Hierarchical**

  If a neural network has a hierarchical structure, the neurons of a lower layer may only communicate with neurons on the next level of layer.
- **Resonance**

  The layers have bi-directional connections, and they can continue sending messages across the connections a number of times until a certain condition is achieved.

2.8.2.2 Intra-layer connections

In more complex structures the neurons communicate among themselves within a layer, this is known as intra-layer connections. There are two types of intra-layer connections.

- **Recurrent**
  the neurons within a layer are fully- or partially connected to one another. After these neurons receive input form another layer, they communicate their outputs with one another a number of times before they are allowed to send their outputs to another layer.

  Generally some conditions among the neurons of the layer should be achieved before they communicate their outputs to another layer.

- **On-center/off surround**
  A neuron within a layer has excitatory connections to itself and its immediate neighbors, and has inhibitory connections to other neurons. One can imagine this type of connection as a competitive gang of neurons. Each gang excites itself and its gang members and inhibits all members of other gangs.

After a few rounds of signal interchange, the neurons with an active output value will win, and is allowed to update its and its gang member's weights.

- (There are two types of connections between two neurons, excitatory or inhibitory. In the excitatory connection, the output of one neuron increases the action potential of the neuron to which it is connected. When the connection type between two neurons is inhibitory, then the output of the neuron sending a message would reduce the activity or action potential of the receiving neuron. One causes the summing mechanism of the next neuron to add while the other causes it to subtract. One excites while the other inhibits.

2.8.3 Learning

The brain basically learns from experience. Neural networks are sometimes called machine-learning algorithms, because changing of its connection weights (training) causes the network to learn the solution to a problem. The strength of connection between the neurons is stored as a weight-value for the specific connection. The system learns new knowledge by adjusting these connection weights. This method is proven highly successful in training of multilayered neural nets.

The network is not just given reinforcement for how it is doing on a task. Information about errors is also filtered back through the system and is used to adjust the connections between the layers, thus improving performance. A form of supervised learning.

The learning ability of a neural network is determined by its architecture and by the algorithmic method chosen for training. The training method usually consists of one of two schemes:

1. **Unsupervised learning**
   The hidden neurons must find a way to organize themselves without help from the outside. In this approach, no sample outputs are provided to the network against which it can measure its predictive performance for a given vector of inputs. This is learning by doing.
2. **Reinforcement learning**
   This method works on reinforcement from the outside. The connections among the neurons in the hidden layer are randomly arranged, then reshuffled as the network is told how close it is to solving the problem.

Reinforcement learning is also called supervised learning, because it requires a teacher. The teacher may be a training set of data or an observer who grades the performance of the network results.

Both unsupervised and reinforcement suffers from relative slowness and inefficiency relying on a random shuffling to find the proper connection weights.

2.8.3.1 Off-line or On-line

One can categorize the learning methods into yet another group, off-line or on-line. When the system uses input data to change its weights to learn the domain knowledge, the system could be in training mode or learning mode.

When the system is being used as a decision aid to make recommendations, it is in the operation mode; this is also sometimes called recall.

- **Off-line**
  In the off-line learning methods, once the systems enters into the operation mode, its weights are fixed and do not change any more. Most of the networks are of the off-line learning type.

- **On-line**
  In on-line or real time learning, when the system is in operating mode (recall), it continues to learn while being used as a decision tool. This type of learning has a more complex design structure.

2.8.3.2 Learning laws

There are a variety of learning laws, which are in common use. These laws are mathematical algorithms used to update the connection weights. Most of these laws are some sorts of variation of the best-known and oldest learning law, Hebb's Rule. Man's understanding of how neural processing actually works is very limited.

Learning is certainly more complex than the simplification represented by the learning laws currently developed. Research into different learning functions continues as new ideas routinely show up in trade publications etc. A few of the major laws are given as an example below.

- **Hebb's Rule**

  The first and the best-known learning Donald Hebb introduced rule. The organization of Behavior in 1949.This basic rule is: If a neuron receives an input from another neuron, and if both are highly active (mathematically have the same sign), the weight between the neurons should be strengthened.

- **Hopfield Law**

  This law is similar to Hebb's Rule with the exception that it specifies the magnitude of the strengthening or weakening.

  It states, "if the desired output and the input are both active or both inactive, increment the connection weight by the learning rate, otherwise decrement the weight by the learning rate.

Most learning functions have some provision for a learning rate, or learning constant. Usually this term is positive and between zero and one.

- **The Delta Rule**

  The Delta Rule is a further variation of Hebb's Rule, and it is one of the most commonly used. This rule is based on the idea of continuously modifying the strengths of the input connections to reduce the difference (the delta) between the desired output value and the actual output of a neuron.

  This rule changes the connection weights in the way that minimizes the mean squared error of the network. The error is back propagated into previous layers one layer at a time. The process of back-propagating the network errors continues until the first layer is reached. The network type called Feed forward, Back-propagation derives its name from this method of computing the error term.

This rule is also referred to as the Windrow-Hoff Learning Rule and the Least Mean Square Learning Rule.

- **Kohonen's Learning Law**

Learning inspired this procedure, developed by Teuvo Kohonen, in biological systems. In this procedure, the neurons compete for the opportunity to learn, or to update their weights. The processing neuron with the largest output is declared the winner and has the capability of inhibiting its competitors as well as exciting its neighbors. Only the winner is permitted output, and only the winner plus its neighbors are allowed to update their connection weights. The Kohonen rule does not require desired output. Therefore it is implemented in the unsupervised methods of learning.

Kohonen has used this rule combined with the on-center/off-surround intra- layer connection to create the self-organizing neural network, which has an unsupervised learning method. On this Internet site by Sue Becker you may see an interactive demonstration of a Kohonen network, which may give you a better understanding. [5]

2.9 How Neural Networks Learn?

Artificial neural networks are typically composed of interconnected "units", which serve as model neurons. The function of the *synapse* is modeled by a modifiable weight, which is associated with each connection. Each unit converts the pattern of incoming activities that it receives into a single outgoing activity that it broadcasts to other units. It performs this conversion in two stages:

1. It multiplies each incoming activity by the weight on the connection and adds together all these weighted inputs to get a quantity called the *total input*.
2. A unit uses an input-output function that transforms the total input into the outgoing activity.

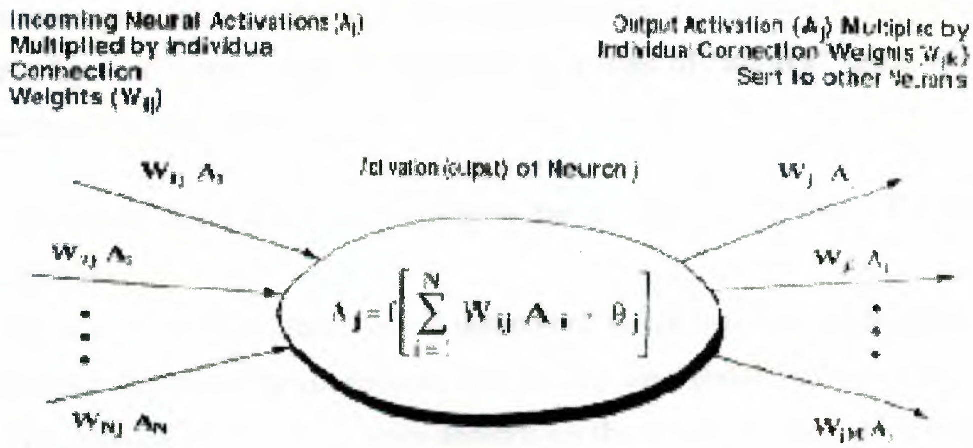


Figure 2.7. The weights and the input-output function (transfer function)

The behavior of an ANN (Artificial Neural Network) depends on both the weights and the input-output function (transfer function) that is specified for the units.

This function typically falls into one of three categories:

- Linear
- Threshold
- Sigmoid

For **linear units**, the output activity is proportional to the total weighted output.

For **threshold units**, the output is set at one of two levels, depending on whether the total input is greater than or less than some threshold value.

For **sigmoid units**, the output varies continuously but not linearly as the input changes. Sigmoid units bear a greater resemblance to real neurons than do linear or threshold units, but all three must be considered rough approximations.

To make a neural network that performs some specific task, we must choose how the units are connected to one another, and we must set the weights on the connections appropriately. The connections determine whether it is possible for one unit to influence another. The weights specify the strength of the influence.

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

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

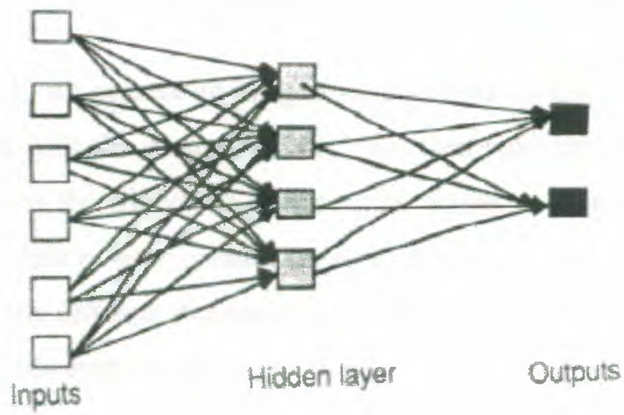


Figure 2.8. Simple type of network

This simple type of network is interesting because the hidden units are free to construct their own representations of the input.

The weights between the input and hidden units determine when each hidden unit is active, and so by modifying these weights, a hidden unit can choose what it represents.

We can teach a three-layer network to perform a particular task by using the following procedure:

1. We present the network with training examples, which consist of a pattern of activities for the input units together with the desired pattern of activities for the output units.
2. We determine how closely the actual output of the network matches the desired output.

3. We change the weight of each connection so that the network produces a better approximation of the desired output.

An Example to illustrate the above teaching procedure:

Assume that we want a network to recognize hand-written digits. We might use an array of, say, 256 sensors, each recording the presence or absence of ink in a small area of a single digit. The network would therefore need 256 input units (one for each sensor), 10 output units (one for each kind of digit) and a number of hidden units.

For each kind of digit recorded by the sensors, the network should produce high activity in the appropriate output unit and low activity in the other output units.

To train the network, we present an image of a digit and compare the actual activity of the 10 output units with the desired activity. We then calculate the error, which is defined as the square of the difference between the actual and the desired activities. Next we change the weight of each connection so as to reduce the error. We repeat this training process for many different images of each different images of each kind of digit until the network classifies every image correctly.

To implement this procedure we need to calculate the error derivative for the weight (EW) in order to change the weight by an amount that is proportional to the rate at which the error changes as the weight is changed. One way to calculate the EW is to perturb a weight slightly and observe how the error changes. But that method is inefficient because it requires a separate perturbation for each of the many weights. Another way to calculate the EW is to use the Back-propagation algorithm which is described below, and has become nowadays one of the most important tools for training neural networks.

2.10 Building A Neural Network

Since 1958, when psychologist Frank Rosenblatt proposed the "Perceptron," a pattern recognition device with learning capabilities, the hierarchical neural network has been the most widely studied form of network structure. A hierarchical neural network is one that links multiple neurons together hierarchically.

The special characteristic of this type of network is its simple dynamics. That is, when a signal is input into the input layer, it is propagated to the next layer by the interconnections between the neurons.

Simple processing is performed on this signal by the neurons of the receiving layer prior to its being propagated on to the next layer. This process is repeated until the signal reaches the output layer completing the processing process for that signal.

The manner in which the various neurons in the intermediary (hidden) layers process the input signal will determine the kind of output signal it becomes (how it is transformed).

As you can see, then, hierarchical network dynamics are determined by the weight and threshold parameters of each of their units. If input signals can be transformed to the proper output signals by adjusting these values (parameters), then hierarchical networks can be used effectively to perform information processing.

Since it is difficult to accurately determine multiple parameter values, a learning method is employed. This involves creating a network that randomly determines parameter values. This network is then used to carry out input-to-output transformations for actual problems. The correct final parameters are obtained by properly modifying the parameters in accordance with the errors that the network makes in the process. Quite a few such learning methods have been proposed. Probably the most representative of these is the error back-propagation learning method proposed by D. E. Rumelhart et al. in 1986. This learning method has played a major role in the recent neurocomputing boom.

The back-propagation paradigm has been tested in numerous applications including bond rating, mortgage application evaluation, protein structure determination, backgammon playing, and handwritten digit recognition. Choosing the right methodology, or back propagation algorithm, is another important consideration.

In working with the financial applications, many have found that the back-propagation algorithm can be very slow. Without using advanced learning techniques to speed the process up, it is hard to effectively apply back propagation to real-world problems.

Over fitting of a neural network model is another area, which can cause beginners difficulty. Over fitting happens when an ANN model is trained on one set of data, and it learns that data too well. This may cause the model to have poor generalization abilities - the model may instead give quite poor results for other sets of data.

For an in-depth coverage of other neural network models and their learning algorithms, please refer to the Technical Reading at the end of this User's Guide, the Technical Reference (sold separately), those papers listed in the Reference, or any other reference books on neural networks and relevant technology. [2]

2.11 A Learning Process

For decades there have been attempts to create computer programs that can *learn* like people - Artificial Intelligence. For example, how do you teach a child to recognize what a chair is? You show him examples telling him "This is a chair ; That one is not a chair" until the child learns the concept of what a chair is. In this stage, the child can look at the examples we have shown him and answer correctly to the question "Is this object a chair?". Furthermore, if we show to the child new objects, that he didn't see before, we could expect him to recognize correctly whether the new object is a chair or not, providing that we've given him enough positive and negative examples. This is exactly the idea behind the perceptron.

2.12 The Perceptron

The perceptron is a program that learns **concepts**, i.e. it can learn to respond with *True* (1) or *False* (0) for inputs we present to it, by repeatedly "studying" examples presented to it.

The Perceptron is a single layer neural network whose weights and biases could be trained to produce a correct target vector when presented with the corresponding input vector. The training technique used is called *the perceptron-learning rule*.

The perceptron generated great interest due to its ability to **generalize** from its training vectors and work with randomly distributed connections. Perceptrons are especially suited for simple problems in pattern classification. The perceptron looks like:

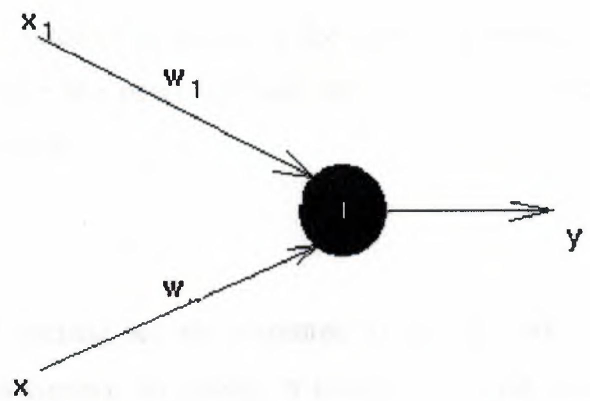


Figure 2.9. Single perceptron

Our perceptron network consists of a single neuron connected to two inputs through a set of 2 weights, with an additional bias input.

The perceptron calculates its output using the following equation:

$$P * W + b > 0$$

Where P is the input vector presented to the network, W is the vector of weights and b is the bias. [3]

2.12.1 The Learning Rule

The perceptron is trained to respond to each input vector with a corresponding target output of either 0 or 1. The learning rule has been proven to converge on a solution in finite time if a solution exists.

The learning rule can be summarized in the following two equations:

For all i:

$$W(i) = W(i) + [T - A] * P(i)$$

$$b = b + [T - A]$$

Where W is the vector of weights, P is the input vector presented to the network, T is the correct result that the neuron should have shown, A is the actual output of the neuron, and b is the bias.

2.12.2 Training

Vectors from a training set are presented to the network one after another. If the network's output is correct, no change is made. Otherwise, the weights and biases are updated using the perceptron-learning rule. An entire pass through all of the input training vectors is called an *epoch*. When such an entire pass of the training set has occurred without error, training is complete. At this time any input training vector may be presented to the network and it will respond with the correct output vector.

If a vector P not in the training set is presented to the network, the network will tend to exhibit *generalization* by responding with an output similar to target vectors for input vectors close to the previously unseen input vector P.

2.12.3 Limitations

Perceptron networks have several limitations. First, the output values of a perceptron can take on only one of two values (True or False). Second, perceptrons can only classify **linearly separable** sets of vectors. If a straight line or plane can be drawn to separate the input vectors into their correct categories, the input vectors are linearly separable and the perceptron will find the solution. If the vectors are not linearly separable learning will never reach a point where all vectors are classified properly.

The most famous example of the perceptron's inability to solve problems with linearly no separable vectors is the Boolean exclusive-or problem.

2.12.4 Developments from the simple perceptron:

Back-Propagated Delta Rule Networks (BP) (sometimes known and multi-layer perceptrons (MLPs)) and Radial Basis Function Networks (RBF) are both well-known developments of the Delta rule for single layer networks (itself a development of the Perceptron Learning Rule). Both can learn arbitrary mappings or classifications. Further, the inputs (and outputs) can have real values. [3]

2.12.5 Implementation

We implemented a single neuron perceptron with 2 inputs. The input for the neuron can be taken from a graphic user interface, by clicking on points in a board. A click with the left mouse button generates a '+' sign on the board, marking that it's a point where the perceptron should respond with 'True'. A click with the right mouse button generates a '-' sign on the board, marking that it's a point where the perceptron should respond with 'False'. When enough points have been entered, the user can click on 'Start', which will introduce these points as inputs to the perceptron, have it learn these input vectors and show a line which corresponds to the linear division of the plane into regions of opposite neuron response. [3]

2.13 A description of the Back Propagation Algorithm

To train a neural network to perform some task, we must adjust the weights of each unit in such a way that the error between the desired output and the actual output is reduced. This process requires that the neural network compute the error derivative of the weights (**EW**). In other words, it must calculate how the error changes as each weight is increased or decreased slightly. The back propagation algorithm is the most widely used method for determining the **EW**.The back-propagation algorithm is easiest to understand if all the units in the network are linear.

The algorithm computes each **EW** by first computing the **EA**, the rate at which the error changes as the activity level of a unit is changed. For output units, the **EA** is simply the difference between the actual and the desired output.

To compute the **EA** for a hidden unit in the layer just before the output layer, we first identify all the weights between that hidden unit and the output units to which it is connected. We then multiply those weights by the **EA**s of those output units and add the products. This sum equals the **EA** for the chosen hidden unit. After calculating all the **EA**s in the hidden layer just before the output layer, we can compute in like fashion the **EA**s for other layers, moving from layer to layer in a direction opposite to the way activities propagate through the network. This is what gives back propagation its name. Once the **EA** has been computed for a unit, it is straightforward to compute the **EW** for each incoming connection of the unit. The **EW** is the product of the EA and the activity through the incoming connection. Note that for non-linear units, the back-propagation algorithm includes an extra step. Before back propagating, the **EA** must be converted into the **EI**, the rate at which the error changes as the total input received by a unit is changed. [10]

A Back-Propagation Network Example

In this example a back-propagation network would be used to solve a specific problem, that one of an X-OR logic gate. That means that patterns have (1,1) should produce a value close to zero in the output node, and input patterns of (1,0) or (0,1) should produce a value near one in the output node. Finding a set of connection weights for this task is not easy; it requires application of the back-propagation algorithm for several thousand iterations to achieve a good set of connection weights and neuron thresholds.

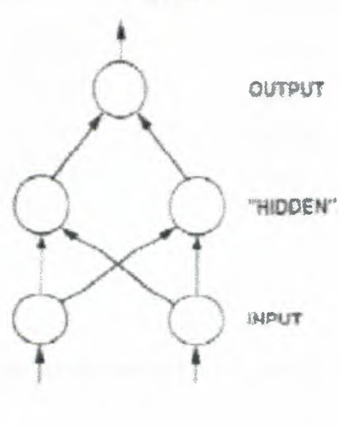


Figure 2.10. The basic architecture of neural network

The basic architecture for this problem has two input nodes, two hidden nodes, and a single output node as shown above.

This structure has variable thresholds on the two hidden and one output node (unit). This means that there are a total of 9 variables in the system:

- 4 weights connecting the input to the hidden nodes
- 2 weights connecting the hidden to the output node
- 3 thresholds.

Suppose we put in a pattern, say (0,1). That mean that there is 0 activation in the left-hand neuron on the first layer and an activation of 1 in the neuron on the right.

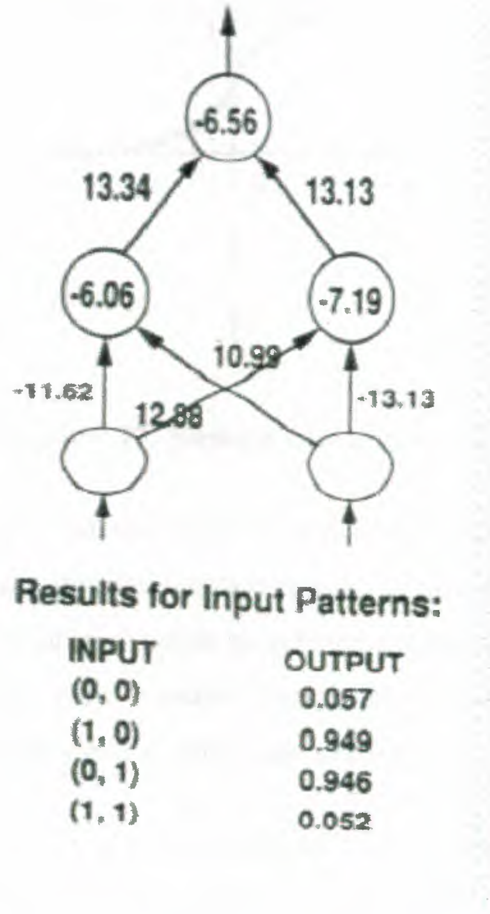


Figure 2.11.The basic architecture of neural network with values

Now we move our attention to the next layer up. For each neuron in this layer, we calculate an input, which is the weighted sum of all the activations from the first layer.

The weighted sum is achieved by vector multiplying the activations in the first layer by a "connection matrix".

In our case we get a value of 0*(-11,62)+ 1*(10,99) = 10,99 for the neuron on the left in the second layer, and 0*(12,88) +1*(13,13) = =13,13 for the neuron on the right.

we add a "threshold" value (which is found for each neuron using the back-propagation rule), and apply an input-output (transfer) function.

The transfer function is defined for each different network. In our case it is a sigmoid:

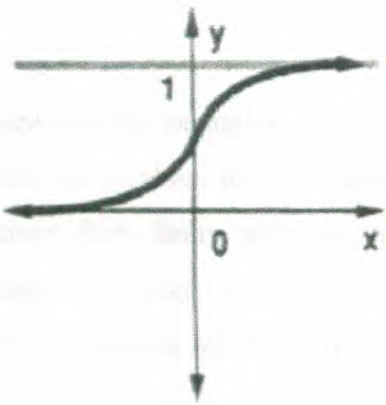


Figure 2.12. Sigmoid function curve

In this case it has been shown, that the activation of the neuron on the left side of the hidden (middle) layer is the transfer function applied to the difference (10,99-6,06) = 4,94. Applying the transfer function yields an activation value close to 1. The activation of the neuron on the right is the transfer function applied to (-13,13+7,19) = -5,14. Applying the transfer function yields a value close to 0. [10]

Approximating the next step, we use a value of 1 for the activation of the neuron on the left, and 0 for the neuron on the right, multiply each activation by its appropriate connection weight, and sum the values as input to the topmost neuron. This is approximately 1*(13,34)+0*(13,13) = 13,34. We add the threshold of -6,56 to obtain a value of 6,78. Applying the transfer function to it will yield a value close to 1 (0,946), which is the desired result. Using the other 3 binary input patterns, we can similarly show that this network yields the desired classification within an acceptable tolerance.

CHAPTER THREE
IMPLEMENTATIONS
OF
NEURAL NETWORKS

3.1 Overview

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

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

3.2 How Brain Maker Neural Networks work?

Neural networks are named after the cells in the human brain that perform intelligent operations. The brain is made up of billions of neuron cells. Each of these cells is like a tiny computer with extremely limited capabilities; however, connected together, these cells form the most intelligent system known. Neural networks are formed from hundreds or thousands of simulated neurons connected together in much the same way as the brain's neurons. Just like people, neural networks learn from experience, not from programming. Neural networks are good at pattern recognition, generalization, and trend prediction. They are fast, tolerant of imperfect data, and do not need formulas or rules. [5]

Neural networks are trained by repeatedly presenting examples to the network. Each example includes both inputs (information you would use to make a decision) and outputs (the resulting decision, prediction, or response). Your network tries to learn each of your examples in turn, calculating its output based on the inputs you provided.

If the network output doesn't match the target output, BrainMaker corrects the network by changing its internal connections. This trial-and-error process continues until the network reaches your specified level of accuracy. Once the network is trained and tested, you can give it new input information, and it will produce a prediction. Designing your neural network is largely a matter of identifying which data is input, and what you want to predict, assess, classify, or recognize.

3.3 Why is it useful?

Neural nets 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 net learns by being given a *training set*. Due to its complex, non-linear structure, the neural net can find relationships in data those humans as unable to do.

3.4 Why doesn't it work all the time?

Neural nets can only learn if the training set consists of good examples. The old saying of *'garbage in- garbage out'* is doubly true for neural nets. Great care should be taken to present decor related 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 net cannot learn from just a few examples.

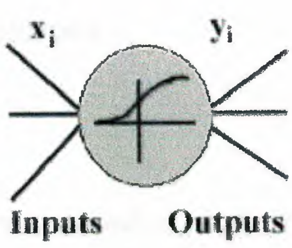


Figure 3.1.The neurons with input, output

3.5 What Applications Should Neural Networks Be Used For?

Neural networks are universal approximates, and they work best if the system you are using them to model has a high tolerance to error. One would therefore not be advised to use a neural network to balance one's checkbook! However they work very well for:
capturing associations or discovering regularities within a set of patterns; where the volume, number of variables or diversity of the data is very great; the relationships between variables are vaguely understood; or, the relationships are difficult to describe adequately with conventional approaches.

3.6 What Are Their Advantages Over Conventional Techniques?

Depending on the nature of the application and the strength of the internal data patterns you can generally expect a network to train quite well. This applies to problems where the relationships may be quite dynamic or non-linear.

ANNs provide an analytical alternative to conventional techniques, which are often limited by strict assumptions of normality, linearity, variable independence etc.

Because an ANN can capture many kinds of relationships it allows the user to quickly and relatively easily model phenomena which otherwise may have been very difficult or impossible to explain otherwise.

3.7 Neural Network Applications

3.7.1 Aerospace

- High performance aircraft autopilot, flight path simulation, aircraft control systems, autopilot enhancements, aircraft component simulation, aircraft component fault detection.

3.7.2 Automotive

- Automobile automatic guidance system, warranty activity analysis.

3.7.3 Banking

- Check and other document reading, credit application evaluation.

3.7.4 Credit Card Activity Checking

- Neural networks are used to spot unusual credit card activity that might possibly be associated with loss of a credit card.

3.7.5 Defense

Weapon steering, target tracking, object discrimination, facial recognition, new kinds of sensors, sonar, radar and image signal processing including data compression, feature extraction and noise suppression, signal/image identification.

3.7.6 Electronics

- Code sequence prediction, integrated circuit chip layout, process control, chip failure analysis, machine vision, voice synthesis, nonlinear modeling.

3.7.7 Entertainment

- Animation, special effects, market forecasting.

3.7.8 Financial

- Real estate appraisals, loan advisor, mortgage screening, corporate bond rating, credit-line use analysis, portfolio trading program, corporate financial analysis, and currency price prediction.

3.7.9 Industrial

- Neural networks are being trained to predict the output gasses of furnaces and other industrial processes. They then replace complex and costly equipment used for this purpose in the past.

3.7.10 Insurance

- Policy application evaluation, product optimization.

3.7.11 Manufacturing

- Manufacturing process control, product design and analysis, process and machine diagnosis, real-time particle identification, visual quality inspection systems, beer testing, welding quality analysis, paper quality prediction, computer-chip quality analysis, analysis of grinding operations, chemical product design analysis, machine maintenance analysis, project bidding, planning and management, dynamic modeling of chemical process system.

3.7.12 Medical

- Breast cancer cell analysis, EEG and ECG analysis, prosthesis design, optimization of transplant times, hospital expense reduction, hospital quality improvement, and emergency-room test advisement.

3.7.13 Oil and Gas

- Exploration

3.7.14 Robotics

- Trajectory control, forklift robot, manipulator controllers, vision systems

3.7.15 Speech

- Speech recognition, speech compression, vowel classification, text-to-speech synthesis.

3.7.16 Securities

- Market analysis, automatic bond rating, stock trading advisory systems.

3.7.17 Telecommunications

Image and data compression, automated information services, real-time translation of spoken language, customer payment processing systems.

3.7.18 Transportation

Truck brake diagnosis systems, vehicle scheduling, routing systems.[6]

3.8 Stocks, Commodities Applications

3.8.1 Neural Networks and Technical Analysis of Currencies

Mr. James O'Sullivan, of O'Sullivan Brothers Investments, Ltd. (Connecticut) has been successfully using many BrainMaker (California Scientific Software) neural networks on a daily basis for three years to do financial forecasting. [7]

He earned $250,000 in one month using neural networks to advise him on his New York Stock Exchange seat trades. Some of his networks are 88-90% accurate in their predictions, according to Mr. O'Sullivan. He uses an automated neural network system that monitors more than twenty different financial markets on a daily basis.

Mr. O'Sullivan has some unusual designs which act more as detectors of specific market conditions, rather than as exact price predictors. He combines the neural network data with other data from his technical analysis software to produce an automated report about a certain market. He gets his data live via satellite from Data Broadcasting and puts it into a charting and technical data module. He has pre-programmed the various algebraic manipulations to be performed on his data before BrainMaker files are built. He does moving averages, changes from period to period, and a few proprietary operations.

He runs new data through the system and produces the one-page report in about thirty seconds. He says at least 80% of his decision-making is based on neural network predictions.

Mr. O'Sullivan has not fully disclosed his neural network designs to us, but his basic insights are still quite valuable. The key is to ask the neural network the right kind of questions.

He asks questions such as "What is the probability of the product (or market) going up 0.618 standard deviations?" and "What percentage of the time does it go up that much?" He also asks questions about the directional behavior of the market and at what price the product (or market) is likely to take off in one direction or the other.

Mr. O'Sullivan's neural networks output several different things such as predicted prices, limits, and directional thrust. One neural network outputs the probability of a certain price occurring the next time period. Another neural network produces best stop price and best target price for long and short positions. Other neural networks produce directional indicators for three market energies. Another predicts the level at which the market is likely to take off.

In one design, the network is given various market conditions as input. The training output is the likelihood of various changes in price. For example, his neural network is told during training that, given similar market conditions, the closing price goes up at least 0.313 standard deviations above the prior day 90% of the time, at least 0.618 deviations 80% of the time, and at least 1 deviation 70% of the time.

An interesting phenomenon of the market is that when a change starts occurring in one direction or the other, there is a point at which it is very likely to continue moving in that direction for several time periods. Once a price reaches that level, there is a reduced risk to buy or sell (whichever is appropriate). Mr. O'Sullivan calls the network that predicts this price level his Risk Barometer network. He uses all the neural networks trained for a specific market when making decisions. For example, if the long-term trend is up, the Fast Movement network is a large positive number, his Risk Barometer says 233.092, and the NYSE is at 250, it could indicate an overreacted market that will reverse itself soon [17].

3.8.2 Predict Bond Prices with Neural Network software

G. R. Pugh & Company has been using a BrainMaker neural network trained on three-to-four years of historical data with an XT-compatible PC to help forecast the next year's corporate bond prices and ratings of 115 public utilities companies. "An XT is more than sufficient; it's a fast program," company president Mr. George Pugh notes.

Learning to use the program and create a neural network from scratch took only two days. The network trained itself to predict bond prices in about four hours.

G. R. Pugh & Company does consulting to predict bond prices for the public utility industry.

He maintains databases with financial and business information on the companies advises with business forecasts and credit risk assessments and predicts the financial and operating health of these companies. His expertise is also used by the brokerage industry. He advises clients by forecasting on the selection of good corporate bonds. His clients need to know more accurately which bonds represent good investments for their customers. Both increases and decreases in bond value provide the potential for profitable investment. Mr. Pugh announced that predicting bond prices with BrainMaker neural network software has been more successful than discriminate analysis and forecasting methods he has used, and even a little better than a person could do. "Discriminate analysis methods are good for getting the direction of lively issues, but neural networks pick up the subtle interactions much better," he explains. The network categorizes the ratings with 100% accuracy within a broad category and 95% accuracy within a subcategory. The mathematical method of discriminate analysis was only 85% accurate within a broad category. (Bonds are rated much like report cards, with broad category ratings such as A, B, C, etc. A subcategory could be A+, for example)." The network makes a significant contribution to his analysis. "The network allows me to pick up things that are not obvious with typical analysis," he says. Moreover, nearly all of the network's difficulties were found to be associated with companies that were experiencing a particularly unusual problem (such as regulatory risk) or had an atypical business relationship (such as being involved in a large sale and lease-back transaction). Ratings also tend to be subjective; financial items are not the only things considered by the rating companies. These influences were not represented in the training facts and this makes predictions difficult. The trained network forecasts next year's Standard & Poor's and Moody's corporate bond ratings (both are industry standards) from the previous year's S & P and Moody's ratings and 23 other measures of each company's financial strength, such as income, sales, returns on equity, five-year growth in sales, and measures of investment, construction, and debt load.

Each of these factors is assigned its own input neuron, and each company's ratings for next year are the outputs of the network. [29]

3.8.3 Predict the S&P 500 Index with Neural Network Software

A highly rated investment firm (Clearwater, FL) manages more than 60 million dollars in investments. LBS rely almost exclusively on computer techniques to guide its decisions in predicting the S&P 500 Index. The firm is a forerunner in using neural network software to recognize patterns and predict indexes and trends for financial forecasting.

The latest approach in forecasting used by LBS integrates an expert system with a BrainMaker neural network to make the most efficient use of the talents of each. The expert system provides rules, which govern the application of the neural network to the prediction. For example, if the expert system says the market is trending and the neural network forecasts the S&P will go up, then a buy signal is generated.

In predicting the S&P 500 index the neural network is trained with only recent market data (less than five years' worth) because it was found that the actual behavior of the market 25 years ago was not the same as it is today. Commonly available indicators are used such as the ADX, MACD, stochastic, DOW, volume, etc. The BrainMaker neural network "window" was found to be most effective at five market days for predicting the S&P 500 index. It was speculated that every weekday might have a certain "tone" to it, so that all Mondays tend to behave similarly. By presenting five-day intervals as historical input data while outputting forecasts for five days in advance, the neural network deals only with the same weekday for each prediction. [29]

The neural network trained by LBS predicts the S&P 500 with an average accuracy of 95%. This statistic was obtained by testing the network on hundreds of days it had never seen before. The network is retrained every night with the most current information to keep its behavior in agreement with the current behavior of the market [18].

3.8.4 Predicting Stock Prices using Neural Network Software

Warren Buffett is a pillar of the financial world, and with good reason. He has parlayed his theories on investing and market analysis into a substantial fortune, while others have used his advice to build their own highly successful investment portfolios. Some, too, have crunched Buffett's investment formulas, or something like them, into a suite of computer programs that produce an electronic version of the Buffett genius. [29]

Walkrich Investment Advisors, a consulting firm out of Cape Girardeau, Missouri, uses BrainMaker Neural Networks to do just that -- produce an investment tool (WRRAT) based loosely on Buffett's ideas and BrainMaker neural networks in predicting stock prices. How well does WRRAT perform in stock price prediction? From January '95 to January '96, a portfolio made up of WRRAT's 20 most under priced stocks would have seen an average advance of 32.63%, compared to the S&P's 31.93% gain over the same period. More recently, WRRAT's 20 most under priced stocks have enjoyed a 44.40% gain from January '96 to February '96, compared to the S&P's 38.65%.

How does WRRAT's forecasting compare to the flesh and blood Buffett? From July '95 to February '96, shares in Berkshire Hathaway, Buffett's holding company, have gained an average of 28%. With WRRAT's 1995 average advance of 32.63%, in financial forecasting Walkrich and BrainMaker can compete with the best.

Walkrich uses a BrainMaker neural network to determine the average premium (discount) the market is currently allocating to particular industries, and then uses that standard in an industry-by-industry neural network analysis designed to determine which stocks are trading below their market value. In that analysis, the neural network will appraise each stock, giving a price estimate (based on price/earnings, price/book and dividend yield), which is adjusted for size, industry, exchange listing and institutional influence.

The neural net's per-stock price estimate is then compared to the corresponding industry average, producing a calculated measure of each stock's relative value -- in short, whether the stock is being under priced or overpriced by the market.

NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

APPLICATIONS OF NEURAL NETWORKS

Graduation Project
COM- 400

Student: Ahmed Qatanani

Supervisor: Assoc.Prof. Dr Adnan Khashman

Nicosia - 2001

ACKNOWLEDGEMENT

First of all I would like to thank Allah Almighty who gave me the strength to complete this project.

Then I would like to thank my Supervisor Assoc.Prof.Dr. Adnan Khashman who is not only a teacher but also a very appreciating personality.

Although I face many difficulties finding appropriate data regarding some topics of the project, but he (Dr.Adnan) has guided me with keen interest and helped me by all means. Dr.Adnan, thanks for your continual support.

I would like to thank all our teachers in the Near East University, Student Advisors and all the Staff of the Faculty of Engineering for giving us the opportunity to experience this remarkable Institute that have showed us preliminary steps towards our professional carrier.

Many thanks to my parents for their continuous moral support and encouragement.

Also many thanks to my friends especially Ahmed El taha, Ahmed wadi, Mohammed Abu shaban and both of my cousins Ata and Ashraf.

ABSTRACT

Technology of Neural Network gives a computer system an amazing capacity to actually learn from input data. Artificial neural networks have provided solutions to problems normally requiring human observation and thought processes. Neural network simulations appear to be a recent development. However, this field was established before the advent of computers, and has survived at least one major setback and several eras. The computing world has a lot to gain fron neural networks. Their ability to learn by example makes them very flexible and powerful.

The most basic components of neural networks are modeled after the structure of the brain. Some neural network structures are not closely to the brain and some does not have a biological counterpart in the brain. However, neural networks have a strong similarity to the biological brain and therefore a great deal of the terminology is borrowed from neuroscience. Given this description of neural networks and how they work, what real world applications are they suited for? Neural networks have broad applicability to real world business problems. In fact, they have already been successfully applied in many industries, science, medicine, Manufacturing, and Sports...etc.

Neural networks are trained by repeatedly presenting examples to the network. Each example includes both inputs and outputs. The network tries to learn each of your examples in turn, calculating its output based on the inputs you provided. Neural networks represent a new technology with many potential uses. Their capabilities have already been proven in a variety of business applications. Almost any neural network application would fit into one business area or financial analysis. There is some potential for using neural networks for business purposes, including resource allocation and scheduling.

INTRODUCTION

Chapter one aimed to show the background of the neural networks and what the reasons and the benefits of using neural networks. And also how the neural networks improved to be new technology in the present and the future. Neural network is defined as massively parallel-distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects: (1) Knowledge is acquired by the network through a learning process, and (2) Interneuron connection strengths known as synaptic weights are used to store the knowledge.

Neural network simulations appear to be a recent development. However, this field was established before the advent of computers, and has survived at least one major setback and several eras. Many important advances have been boosted by the use of inexpensive computer emulations. Neural networks learn by example. They cannot be programmed to perform a specific task.

The development of true Neural Networks is a fairly recent event, which has been met with success. The future of Neural Networks is wide open, and may lead to many answers and/or questions.

Chapter two describes the architectures and basic components of neural networks; the most basic components of neural networks are modeled after the structure of the brain. Some neural network structures are not closely to the brain and some does not have a biological counterpart in the brain. However, neural networks have a strong similarity to the biological brain and therefore a great deal of the terminology is borrowed from neuroscience. Neural networks are named after the cells in the human brain that perform intelligent operations. The brain is made up of billions of neuron cells. Each of these cells is like a tiny computer with extremely limited capabilities; however, connected together, these cells form the most intelligent system known. Neural networks are formed from hundreds or thousands of simulated neurons connected together in much the same way as the brain's neurons.

The term `architecture' has been much abused in the history of mankind. It has many meanings depending on whether you are talking about buildings, inside of computers or neural networks among other things. Even in neural networks, the term architecture and what we have been referring to as `type' of neural network are used interchangeably. So when we refer to such and such architecture, it means the set of possible interconnections (also called as topology of the network) and the learning algorithm defined for it.

Also Learning algorithms, which were considered for a single perceptron, linear adaline, and multilayer perceptron belong to the class of supervised learning algorithms. Two basic groups of unsupervised learning algorithms and related self-organizing neural networks, namely: Hebbian Learning, Competitive Learning.

Networks such as the one just described are called artificial neural networks (ANNs), in the sense that they represent simplified models of natural nerve or neural networks.

Chapter three shows the fields where the Neural Networks can be applied, Neural Networks are performing successfully where other methods do not, recognizing and matching complicated, vague, or incomplete patterns

Neural networks have been applied in solving a wide variety of problems. The most common use for neural networks is to project what will most likely happen. There are many applications of Neural Networks that can be applied in the real world. Although one may apply neural network systems for prediction, diagnosis, planning, monitoring, debugging, repair, instruction, and control, the most successful applications of neural networks are in categorization and pattern recognition.

A number of real applications can also be found in the NeuroForecaster package. Based on these successful applications, it is therefore evident that the neural network technology can be applied to many real-world problems especially those related to business, financial and engineering modeling.

Chapter four highlights the important application of Neural Network in business, management and finance, the application would fit into one business area or financial analysis. There is some potential for using neural networks for business purposes, including resource allocation and scheduling. There is also a strong potential for using neural networks for database mining, that is, searching for patterns implicit within the explicitly stored information in databases. Most of the funded work in this area is classified as proprietary. Thus, it is not possible to report on the full extent of the work going on. Most work is applying neural networks, such as the Hopfield-Tank network for optimization and scheduling.

CHAPTER ONE
BACKGROUND
ON
NEURAL NETWORKS

1.1 Overview

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

It would be the most remarkable thing ever. And all humans can live happily ever after. This is the aim of artificial intelligence in general. Neural network simulations appear to be a recent development. However, this field was established before the advent of computers, and has survived at least one major setback and several eras. Many important advances have been boosted by the use of inexpensive computer emulations. The future of Neural Networks is wide open, and may lead to many answers and/or questions. Is it possible to create a conscious machine? What rights do these computers have? How does the human mind work? What does it mean to be human?

Neural networks process information in a similar way the human brain does. The network is composed of a large number of highly interconnected processing elements (neurons) working in parallel to solve a specific problem. Neural networks learn by example. They cannot be programmed to perform a specific task. The examples must be selected carefully otherwise useful time is wasted or even worse the network might be functioning incorrectly.

1.2 Background of Neural Networks

Many tasks, which seem simple for us, such as reading a handwritten note or recognizing 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 analyzes data by passing it through several simulated processors which are interconnected with synaptic-like "weights". Although the programming and mathematics behind neural network technologies are complex, using neural network software can be quite simple and the results are often quite extraordinary. Once you have collected several records of the data you wish to analyze, the network will run through them and "learn" how the inputs of each record may be related to the result. Each "record" might be a machine on an assembly line, or a particular stock, or the weather one day.

If the record was a patient at a hospital, the record's inputs (such as: age, sex, body fat, allergies, blood pressure) and it's related output (such as: did the drug work in this case?) are both fed into the "neurons" of the network. The network then continually refines itself until it can produce an accurate response when given those particular inputs. After training on a few dozen cases, the network begins to organize 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 internal mapping of weights that can accurately reproduce the expected output. Once you realize 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. [4]

1.3 What is a Neural Network?

There are definitions of Neural Network

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

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

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

Neural networks: is a buzz word. 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 modeling - and are strongly supported by industry.

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

Neural Network: technology gives a computer system an amazing capacity to actually learn from input data. Artificial neural networks have provided solutions to problems normally requiring human observation and thought processes.

Neural network: is powerful computational tools that can be used for classification, pattern recognition, empirical modeling and for many other tasks. Neural networks (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 modeled.

Neural Network: is a system loosely modeled on the human brain. The field goes by many names, such as connectionism; parallel distributed processing, neuron-computing, natural intelligent systems, machine learning algorithms, and artificial neural networks. It is an attempt to simulate within specialized hardware or sophisticated software, the multiple layers of simple processing elements called neurons.

Each neuron is linked to certain of its neighbors with varying coefficients of connectivity that represent the strengths of these connections. Learning is accomplished by adjusting these strengths to cause the overall network to output appropriate results. [13]

1.4 Historical background of Neural Network

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

These pioneers were able to develop convincing technology, which surpassed the limitations identified by Minsky and Papert. Minsky and Papert, published a book (in 1969) in which they summed up a general feeling of frustration (against neural networks) among researchers, and was thus accepted by most without further analysis. Currently, the neural network field enjoys a resurgence of interest and a corresponding increase in funding.

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

1. **First Attempts:** There were some initial simulations using formal logic. McCulloch and Pitts (1943) developed models of neural networks based on their understanding of neurology. These models made several assumptions about how neurons worked. Their networks were based on simple neurons, which were considered to be binary devices with fixed thresholds. The results of their model were simple logic functions such as "a or b" and "a and b". Another attempt was by using computer simulations. Two groups (Farley and Clark, 1954; Rochester, Holland, Haibit and Duda, 1956). The first group (IBM researchers) maintained closed contact with neuroscientists at McGill University. So whenever their models did not work, they consulted the neuroscientists. This interaction established a multidisciplinary trend, which continues to the present day.
2. **Promising & Emerging Technology:** Not only was neuroscience influential in the development of neural networks, but psychologists and engineers also contributed to the progress of neural network simulations. Rosenblatt (1958) stirred considerable interest and activity in the field when he designed and developed the *Perceptron*. The Perceptron had three layers with the middle layer known as the association layer. This system could learn to connect or associate a given input to a random output unit. Another system was the ADALINE (*ADAptive Linear Element*), which was developed in 1960 by Widrow and Hoff (of Stanford University). The ADALINE was an analogue electronic device made from simple components. The method used for learning was different to that of the Perceptron, it employed the Least-Mean-Squares (LMS) learning rule.

3. **Period of Frustration & Disrepute:** In 1969 Minsky and Papert wrote a book in which they generalized the limitations of single layer Perceptrons to multilayered systems. In the book they said: "...our intuitive judgment that the extension (to multilayer systems) is sterile". The significant result of their book was to eliminate funding for research with neural network simulations. The conclusions supported the disenchantment of researchers in the field. As a result, considerable prejudice against this field was activated.
4. **Innovation:** Although public interest and available funding were minimal, several researchers continued working to develop neuromorphically based computational methods for problems such as pattern recognition. During this period several paradigms were generated which modern work continues to enhance. Grossberg's (Steve Grossberg and Gail Carpenter in 1988) influence founded a school of thought, which explores resonating algorithms. They developed the ART (Adaptive Resonance Theory) networks based on biologically plausible models. Anderson and Kohonen developed associative techniques independent of each other.

Klopf (A. Henry Klopf) in 1972, developed a basis for learning in artificial neurons based on a biological principle for neuronal learning called *heterostasis*. Werbos (Paul Werbos 1974) developed and used the *back-propagation* learning method, however several years passed before this approach was popularized. Back-propagation nets are probably the most well known and widely applied of the neural networks today.

In essence, the back-propagation net. Is a Perceptron with multiple layers, a different threshold function in the artificial neuron, and a more robust and capable learning rule? Mari (A. Shun-Ichi 1967) was involved with theoretical developments: he published a paper, which established a mathematical theory for a learning basis (error-correction method) dealing with adaptive pattern classification.

While Fukushima (F. Kunihiko) developed a stepwise trained multilayered neural network for interpretation of handwritten characters. The original network was published in 1975 and was called the *Cognitron*.

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

Today: Significant progress has been made in the field of neural networks-enough to attract a great deal of attention and fund further research.

Advancement beyond current commercial applications appears to be possible, and research is advancing the field on many fronts. Neurally based chips are emerging and applications to complex problems developing. Clearly, today is a period of transition for neural network technology. [4]

1.5 Why use neural networks?

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

This expert can then be used to provide projections given new situations of interest and answer "what if" questions.

Other advantages include:

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

2. Self-Organization: An ANN can create its own organization or representation of the information it receives during learning time.
3. Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
4. Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

1.6 Neural networks versus conventional computers

Neural networks take a different approach to problem solving than that of conventional computers. Conventional computers use an algorithmic approach i.e. the computer follows a set of instructions in order to solve a problem.

Unless the specific steps that the computer needs to follow are known the computer cannot solve the problem. That restricts the problem solving capability of conventional computers to problems that we already understand and know how to solve.

But computers would be so much more useful if they could do things that we don't exactly know how to do.

Neural networks process information in a similar way the human brain does. The network is composed of a large number of highly interconnected processing elements (neurons) working in parallel to solve a specific problem.

Neural networks learn by example. They cannot be programmed to perform a specific task. The examples must be selected carefully otherwise useful time is wasted or even worse the network might be functioning incorrectly.

The disadvantage is that because the network finds out how to solve the problem by itself, its operation can be unpredictable.

On the other hand, conventional computers use a cognitive approach to problem solving; the way the problem is to solve must be known and stated in small unambiguous instructions.

These instructions are then converted to a high-level language program and then into machine code that the computer can understand.

These machines are totally predictable; if anything goes wrong is due to a software or hardware fault. Neural networks and conventional algorithmic computers are not in competition but complement each other. There are tasks are more suited to an algorithmic approach like arithmetic operations and tasks that are more suited to neural networks.

Even more, a large number of tasks require systems that use a combination of the two approaches (normally a conventional computer is used to supervise the neural network) in order to perform at maximum efficiency.

1.7 What are Neural Networks Used For?

Their applications are almost limitless but fall into a few simple categories.

Classification: Customer/Market profiles, medical diagnosis, signature verification, loan risk evaluation, voice recognition, image recognition, spectra identification, property valuation, classification of cell types, microbes, materials, samples.

Forecasting: Future sales, production requirements, market performance, economic indicators, energy requirements, medical outcomes, chemical reaction products, weather, crop forecasts, environmental risk, horse races, jury panels.

Modeling: Process control, systems control, chemical structures, dynamic systems, signal compression, plastics molding, welding control, robot control, and many more. [13]

1.8 Who Needs Neural Network?

People that have to work with or analyze data of any kind. People in business, finance, industry, education and science whose problems are complex, laborious, fuzzy or simply un-resolvable using present methods. People who want better solutions and wish to gain a competitive edge.

- Computer scientists want to find out about the properties of non-symbolic information processing with neural nets and about learning systems in general.
- Statisticians use neural nets as flexible, nonlinear regression and classification models.
- Engineers of many kinds exploit the capabilities of neural networks in many areas, such as signal processing and automatic control.
- Cognitive scientists view neural networks as a possible apparatus to describe models of thinking and consciousness (High-level brain function).
- Neuro-physiologists use neural networks to describe and explore medium-level brain function (e.g. memory, sensory system, motorics).
- Physicists use neural networks to model phenomena in statistical mechanics and for a lot of other tasks.
- Biologists use Neural Networks to interpret nucleotide sequences.

Philosophers and some other people may also be interested in Neural Networks for various reasons.

1.9 Past and Present

The development of true Neural Networks is a fairly recent event, which has been met with success. Two of the different systems (among the many) that have been developed are: the basic feedforward Network and the Hopfield Net. Neural networks are 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 modeling - and are strongly supported by industry. There is a lot of up-to-date information available on the WWW and we have listed important web sites, which will open the doors for you to a very exciting field of research. [13]

1.10 Future of Neural Network

The future of Neural Networks is wide open, and may lead to many answers and/or questions. Is it possible to create a conscious machine? What rights do these computers have? How does the human mind work? What does it mean to be human? Because gazing into the future is somewhat like gazing into a crystal ball, so it is better to quote some "predictions". Each prediction rests on some sort of evidence or established trend which, with extrapolation, clearly takes us into a new realm.

Prediction1: Neural Networks will fascinate user-specific systems for education, information processing, and entertainment. "Alternative realities", produced by comprehensive environments, are attractive in terms of their potential for systems control, education, and entertainment. This is not just a far-out research trend, but is something, which is becoming an increasing part of our daily existence, as witnessed by the growing interest in comprehensive "entertainment centers" in each home. This "programming" would require feedback from the user in order to be effective but simple and "passive" sensors (e.g. fingertip sensors, gloves, or wristbands to sense pulse, blood pressure, skin ionization, and so on), could provide effective feedback into a neural control system. This could be achieved, for example, with sensors that would detect pulse, blood pressure, skin ionization, and other variables, which the system could learn to correlate with a person's response state.

Prediction2: Neural networks, integrated with other artificial intelligence technologies, methods for direct culture of nervous tissue, and other exotic technologies such as genetic engineering, will allow us to develop radical and exotic life-forms whether man, machine, or hybrid.

Prediction3: Neural networks will allow us to explore new realms of human capability realms previously available only with extensive training and personal discipline. So a specific state of consciously induced neurophysiologically observable awareness is necessary in order to facilitate a man machine system interface. [32]

1.11 Are there any limits to Neural Networks?

The major issues of concern today are the scalability problem, testing, verification, and integration of neural network systems into the modern environment. Neural network programs sometimes become unstable when applied to larger problems.

The defense, nuclear and space industries are concerned about the issue of testing and verification. The mathematical theories used to guarantee the performance of an applied neural network are still under development. The solution for the time being may be to train and test these intelligent systems much as we do for humans.

Also there is some more practical problems like: The operational problem encountered when attempting to simulate the parallelism of neural networks. Since the majority of neural networks are simulated on sequential machines, giving rise to a very rapid increase in processing time requirements as size of the problem expands. Solution: implement neural networks directly in hardware, but these need a lot of development still. Instability to explain any results that they obtain. Networks function as "black boxes" whose rules of operation are completely unknown.

1.12 Advantages of Neural Network

An advantage is that the programmer doesn't need to feed the system with expert knowledge about the model. All the network needs to have is some input data along with the preferred output.

1. They deal with the non-linearities in the world in which we live.
2. They handle noisy or missing data.
3. They create their own relationship amongst information - no equations!
4. They can work with large numbers of variables or parameters.
5. They provide general solutions with good predictive accuracy.

1.13 Disadvantages of Neural Network

Most neural networks don't work with probabilities. This implies that the answer given by a system working on a neural network cannot be connected with a given probability. Nor can the system calculate what the second best answer is. For a number of applications this is adequate, but for a decision support system, the probabilities of the answer should be provided. Furthermore alternative answers should be available from the system. Another problem with neural networks is that it is impossible to follow the reasoning behind a given answer.

In a decision support system it is desirable to know the arguments for an answer, so the operator can verify that the choice has been made on sound reasoning from the system. A neural network has to go through a training phase before it can be taken into use. If the programmer cant get lots of examples of the behavior he requires from the system, the system cant be trained properly.

CHAPTER TWO
ARCHITECTURES
OF
NEURAL NETWORKS

2.1 Overview

Neural networks should be applied in situations where traditional techniques have failed to give satisfactory results, or where a small improvement in modeling performance can make a significant difference in operational efficiency or in bottom-line profits. Direct marketing is an excellent example of where a small improvement can lead to significant results. The response rate on direct marketing campaigns is usually quite low. A five percent response rate is often considered very good.

The human nervous system, it is now known, consists of an extremely large number of nerve cells, or neurons, which operate in parallel to process various types of information. Neurocomputing involves processing information by means of changing the states of networks formed by interconnecting extremely large numbers of simple processing elements, which interact with one another by exchanging signals. Networks such as the one just described are called artificial neural networks (ANNs), in the sense that they represent simplified models of natural nerve or neural networks.

An individual wanting to investigate this emerging technology and explore ways in which it can improve his/her organization is advised to consult with neural network practitioners who have experience in developing and implementing models for use in commercial applications. The bottom line is that any manager interested in getting more useful information from available data should consider neural network technology as an option. They can be used by aggressive organizations to focus available resources more effectively, thus gaining a valuable competitive edge.

2.2 Neural Network Computing

The majority of information processing today is carried out by digital computers. This has led to the widely held misperception that information processing is dependent on digital computers.

However, if we look at cybernetics and the other disciplines that form the basis of information science, we see that information processing originates with living creatures in their struggle to survive in their environments, and that the information being processed by computers today accounts for only a small part - the automated portion - of this.

Viewed in this light, we can begin to consider the possibility of information processing devices that differ from conventional computers. In fact, research aimed at realizing a variety of different types of information processing devices is already being carried out, albeit in the shadows of the major successes achieved in the realm of digital computers. One direction that this research is taking is toward the development of an information-processing device that mimics the structures and operating principles found in the information processing systems possessed by humans and other living creatures.

Digital computers developed rapidly in and after the late 1940's, and after originally being applied to the field of mathematical computations, have found expanded applications in a variety of areas, to include text (word), symbol, image and voice processing, i.e. pattern information processing, robot control and artificial intelligence. However, the fundamental structure of digital computers is based on the principle of sequential (serial) processing, which has little if anything in common with the human nervous system. [16]

The human nervous system, it is now known, consists of an extremely large number of nerve cells, or neurons, which operate in parallel to process various types of information. By taking a hint from the structure of the human nervous system, we should be able to build a new type of advanced parallel information processing device.

In addition to the increasingly large volumes of data that we must process as a result of recent developments in sensor technology and the progress of information technology, there is also a growing requirement to simultaneously gather and process huge amounts of data from multiple sensors and other sources.

This situation is creating a need in various fields to switch from conventional computers that process information sequentially; to parallel computers equipped with multiple processing elements aligned to operate in parallel to process information.

Besides the social requirements just cited, a number of other factors have been at work during the 1980's to prompt research on new forms of information processing devices. For instance, recent neuropsychological experiments have shed considerable light on the structure of the brain, and even in fields such as cognitive science, which study human information processing processes at the macro level, we are beginning to see proposals for models that call for multiple processing elements aligned to operate in parallel.

Research in the fields of mathematical science and physics is also concentrating more on the mathematical analysis of systems comprising multiple elements that interact in complex ways. These factors gave birth to a major research trend aimed at clarifying the structures and operating principles inherent in the information processing systems of human beings and other animals, and constructing an information processing device based on these structures and operating principles. The term "neurocomputing" is the name used to refer to the information engineering aspects of this research. [15]

2.3 The Biological Foundation of NeuroComputing

Neurocomputing involves processing information by means of changing the states of networks formed by interconnecting extremely large numbers of simple processing elements, which interact with one another by exchanging signals. Networks such as the one just described are called artificial neural networks (ANNs), in the sense that they represent simplified models of natural nerve or neural networks.

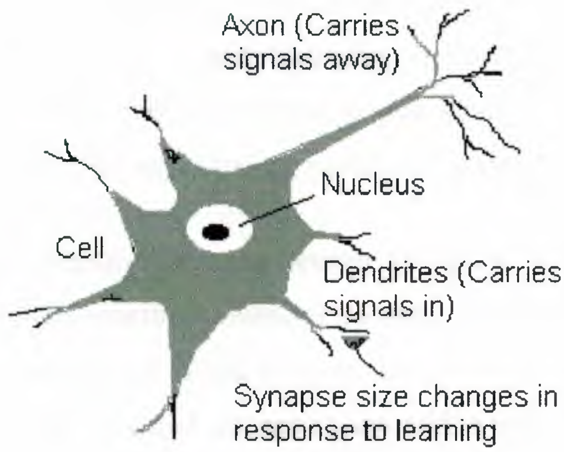


Figure 2.1. A simple neuron cell

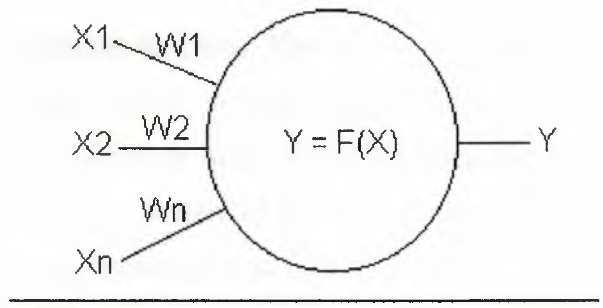


Figure 2.2. A schematic diagram of a neuron

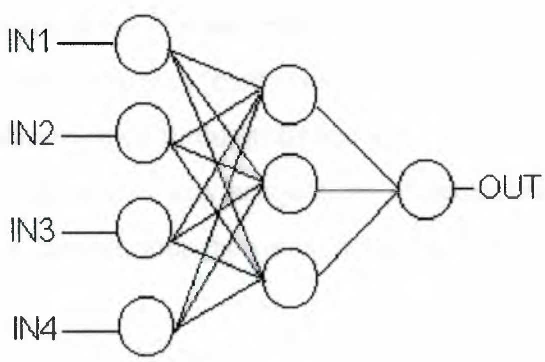


Figure 2.3. A feed forward neural network

The basic processing element in the nervous system is the neuron. The human brain is composed of about 1011 of over 100 types.

Tree-like networks of nerve fiber called dendrites are connected to the cell body or soma, where the cell nucleus is located. Extending from the cell body is a single long fiber called the axon, which eventually branches into strands and sub strands, and are connected to other neurons through synaptic junctions, or synapses.

The transmission of signals from one neuron to another at synapses is a complex chemical process in which specific transmitter substances are released from the sending end of the junction. The effect is to rise to lower the electrical potential inside the body of the receiving cell. If the potential reaches a threshold, a pulse is sent down the axon - we then say the cell has "fired".

In a simplified mathematical model of the neuron, the effects of the synapses are represented by "weights" which modulates the effect of the associated input signals, and the nonlinear characteristics exhibited by neurons is represented by a transfer function which is usually the sigmoid function. The neuron impulse is then computed as the weighted sum of the input signals, transformed by the transfer function. The learning capability of an artificial neuron is achieved by adjusting the weights in accordance to the chosen learning algorithm, usually by a small amount *Wj = **Xj where * is called the learning rate and * the momentum rate. [13]

2.4 Brain plasticity

- At the early stage of the human brain development (the first two years from birth) about 1 million synapses (hard-wired connections) are formed per second.
- Synapses are then modified through the learning process (plasticity of a neuron).
- In an adult brain the above may account for plasticity two mechanisms: creation of new synaptic connections between neurons, and modification of existing synapses.

2.5 What can you do with an NN and what not?

In principle, NNs can compute any computable function, i.e., they can do everything a normal digital computer can do. In practice, NNs are especially useful for classification and function approximation/mapping problems which are tolerant of some imprecision, which have lots of training data available, but to which hard and fast rules (such as those that might be used in an expert system) cannot easily be applied.

Almost any mapping between vector spaces can be approximated to arbitrary precision by feed forward NNs (which are the type most often used in practical applications) if you have enough data and enough computing resources.

To be somewhat more precise, feed forward networks with a single hidden layer, under certain practically-satisfied assumptions are statistically consistent estimators of, among others, arbitrary measurable, square-integral regression functions, binary classifications.

NNs are, at least today, difficult to apply successfully to problems that concern manipulation of symbols and memory. And there are no methods for training NNs that can magically create information that is not contained in the training data.

2.6 Taxonomy of neural networks

From the point of view of their active or decoding phase, artificial neural networks can be classified into feed forward (static) and feedback (dynamic, recurrent) systems.

From the point of view of their learning or encoding phase, artificial neural networks can be classified into supervised and unsupervised systems.

2.6.1 Feed forward supervised networks

These networks are typically used for function approximation tasks. Specific examples include:

- Linear recursive least-mean-square (LMS) networks
- Back propagation networks
- Radial Basis networks.

2.6.2 Feed forward unsupervised networks

These networks are used to extract important properties of the input data and to map input data into a ``representation" domain. Two basic groups of methods belong to this category.

- Hebbian networks performing the Principal Component Analysis of the input data, also known as the Karhunen-Loeve Transform.
- Competitive networks used to performed Learning Vector Quantization, or tessellation of the input data set. Self-Organizing Kohonen Feature Maps also belong to this group. [26]

2.7 The Analogy to the Brain

The most basic components of neural networks are modeled after the structure of the brain. Some neural network structures are not closely to the brain and some does not have a biological counterpart in the brain. However, neural networks have a strong similarity to the biological brain and therefore a great deal of the terminology is borrowed from neuroscience.

2.7.1 The Biological Neuron

The most basic element of the human brain is a specific type of cell, which provides us with the abilities to remember, think, and apply previous experiences to our every action. These cells are known as neurons; each of these neurons can connect with up to 200000 other neurons. The power of the brain comes from the numbers of these basic components and the multiple connections between them. All natural neurons have four basic components, which are dendrites, soma, axon, and synapses. Basically, a biological neuron receives inputs from other sources, combines them in some way, performs a generally nonlinear operation on the result, and then output the final result. The figure below shows a simplified biological neuron and the relationship of its four components.

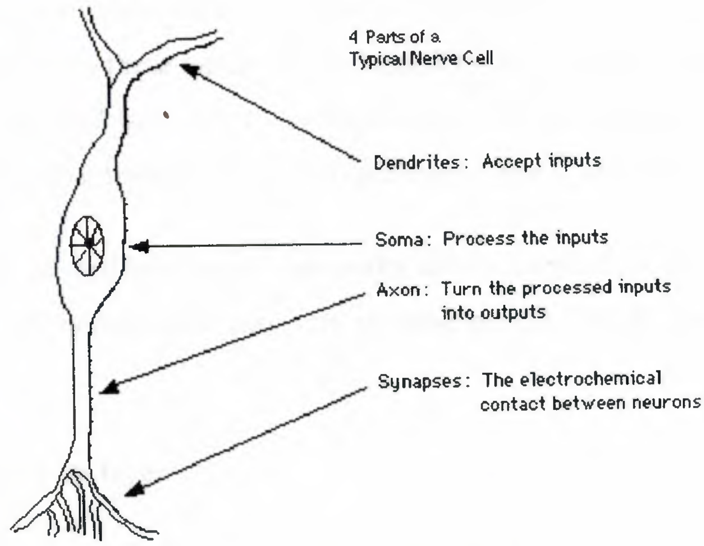


Figure 2.4. A simplified biological neuron

2.7.2 The Artificial Neuron

The basic unit of neural networks, the artificial neurons, simulates the four basic functions of natural neurons. Artificial neurons are much simpler than the biological neuron; the figure below shows the basics of an artificial neuron.

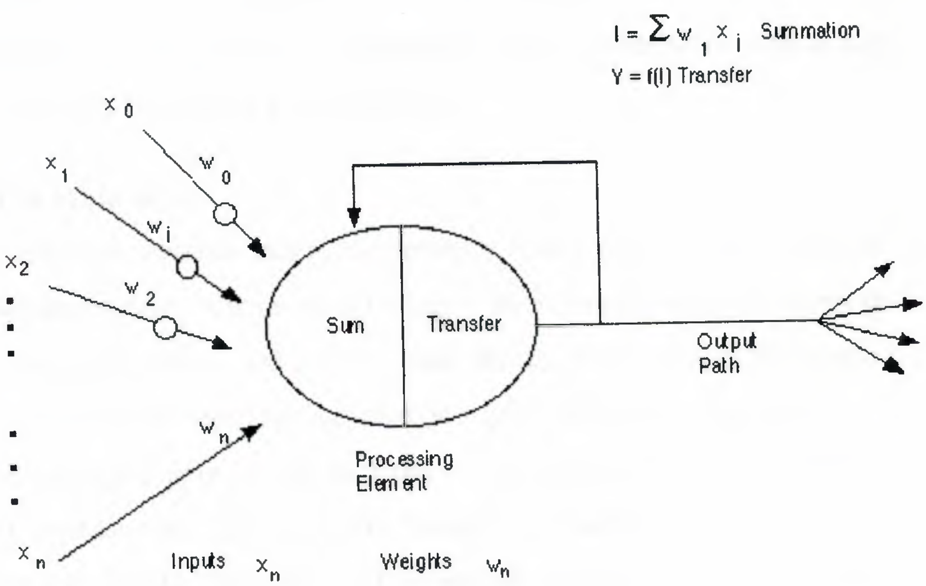


Figure 2.5. The basics of an artificial neuron

Note that various inputs to the network are represented by the mathematical symbol, x (n). Each of these inputs are multiplied by a connection weight, these weights are represented by w (n). In the simplest case, these products are simply summed, fed through a transfer function to generate a result, and then output.

Even though all artificial neural networks are constructed from this basic building block the fundamentals may vary in these building blocks and there are differences.

2.7.3 Neural layer on layer

Neural systems consist of layers of neurons that are connected to each other. Typically, there are three layers: an input layer, an output layer, and a hidden layer. One type of neural system architecture that I have used for financial forecasting is known as a feed forward network with supervised learning.

This type of system has two or more layers, with neurons in one layer receiving information only from the previous layer and sending outputs only to the next layer. Neurons in a given layer do not interconnect. Each neuron in a layer is connected to every neuron of the succeeding layer, with mathematical weights (or connection strengths) assigned to their connections. This is known as "fully connected" network configurations.

2.7.3.1 The input layer

The input layer presents data to the network. The number of data categories determines the number of neurons in the input layer. Each category of input data requires one input neuron, and it is here that the size and structure of the neural system must be determined. For instance, in a S&P 500 or DJIA prediction system, if your input data include each day's closing price for the Deutschemark, S&P 500, Japanese yen, Treasury bills, Eurodollars, Swiss franc, U.S. dollar index, Treasury bonds, DJIA and gold, as well as the discount and Fed funds rates (a total of 12 categories of data), your network would have 12 neurons in the input layer. Massaging the data with moving averages, ratios and so on to eliminate data noise will affect the number of input neurons.

Coupled with each day's input data would be the next day's S&P 500 or DJIA closing price. Each of these input/output pairs of data or training pattern is called "fact."

The hidden layer is composed of neurons that are connected to neurons in the input and output layers but do not connect directly with the outside world. The hidden layer is where the system recodes the input data into a form that captures the hidden correlations, allowing the system to generalize from previously learned facts to new inputs.

Experimentation often determines the number of hidden layers and the appropriate number of neurons in them. Too few neurons impair the network and prevent it from correctly mapping inputs to outputs, while too many neurons impede generalization by allowing the network to "memorize" the patterns presented to it without extracting any of the salient features (similar to curve-fitting or over optimization).

Then, when presented with new patterns, the network cannot process them properly because it has not discovered the underlying relationships.

2.7.3.2 The output layer

Each neuron in the output layer receives its inputs from each neuron in the hidden layer. Your desired output determines how many output neurons the system needs.

Each output category requires one output neuron. Thus, if we want to predict the next day's open, high, low and close for the S&P 500 or the DJIA, the system would, in fact, need four neurons in the output layer. With supervised learning, you would provide the neural system with "facts" that represent input training patterns (today's prices, discount rate and Fed funds rate) that you expect the system to encounter subsequently during trading, and an output-training pattern (next day's prices) that you want it to forecast.

In this manner, during training the system forecasts as its output the next day's S&P 500 or DJIA level, which is then used to adjust each neuron's connection weight, so that during subsequent training iterations, the system will be more likely to forecast the correct output.

For the system to learn during training there must be a way to alter the connection weights in terms of how much and in which direction they will be changed. This algorithm, or paradigm, is known as the "learning law." While numerous learning laws can be applied to neural systems, perhaps the most widely used is the generalized delta rule or back propagation method.

During each iteration of training, the inputs presented to the network generate "a forward flow of activation" from the input to the output layer. Then, whenever the output forecast by the system (next day's S&P 500 or DJIA) is incorrect when compared with its corresponding value in the training pattern, information will flow backward from the output layer to the input layer, adjusting the weights on the inputs along the way. On the next training iteration, when the system is presented with the same input data, it will be more likely to forecast the correct output.

The learning law for a given network defines precisely how to modify these connection weights between neurons to minimize output errors during subsequent training iterations. If no error occurs, then no learning is needed for that fact. Eventually, when the system has completed learning on all of the facts, it reaches a stable state and is ready for further testing.

2.8 Design

The developer must go through a period of trial and error in the design decisions before coming up with a satisfactory design. The design issues in neural networks are complex and are the major concerns of system developers.

Designing a neural network consist of:

- Arranging neurons in various layers.
- Deciding the type of connections among neurons for different layers, as well as among the neurons within a layer.
- Deciding the way a neuron receives input and produces output.
- Determining the strength of connection within the network by allowing the network learns the appropriate values of connection weights by using a training data set.

2.8.1 Layers

Biologically, neural networks are constructed in a three dimensional way from microscopic components. These neurons seem capable of nearly unrestricted interconnections. This is not true in any man-made network. Artificial neural networks are the simple clustering of the primitive artificial neurons. This clustering occurs by creating layers, which are then connected to one another. How these layers connect may also vary. Basically, all artificial neural networks have a similar structure of topology. Some of the neurons interface the real world to receive its inputs and other neurons provide the real world with the network's outputs. All the rest of the neurons are hidden form view.

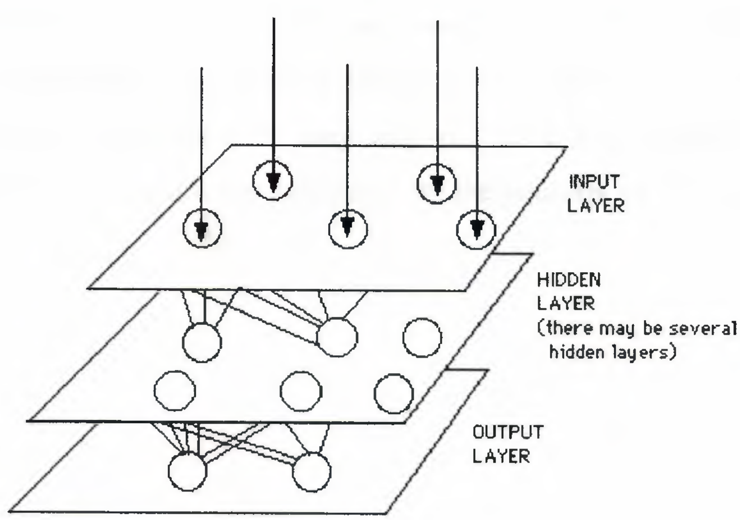


Figure 2.6. Artificial neural network

As the figure above shows, the neurons are grouped into layers. The input layer consists of neurons that receive input form the external environment.

The output layer consists of neurons that communicate the output of the system to the user or external environment. There are usually a number of hidden layers between these two layers; the figure above shows a simple structure with only one hidden layer. When the input layer receives the input its neurons produce output, which becomes input to the other layers of the system. The process continues until a certain condition is satisfied or until the output layer is invoked and fires their output to the external environment. To determine the number of hidden neurons the network should have to perform its best, one are often left out to the method trial and error. If you increase the hidden number of neurons too much you will get an over fit, that is the net will have problem to generalize. The training set of data will be memorized, making the network useless on new data sets.

2.8.2 Communication and types of connections

Neurons are connected via a network of paths carrying the output of one neuron as input to another neuron. These paths is normally unidirectional, there might however be a two-way connection between two neurons, because there may be a path in reverse direction.

A neuron receives input from many neurons, but produce a single output, which is communicated to other neurons. The neuron in a layer may communicate with each other, or they may not have any connections. The neurons of one layer are always connected to the neurons of at least another layer.

2.8.2.1 Inter-layer connections

There are different types of connections used between layers; these connections between layers are called inter-layer connections.

- **Fully connected**

  Each neuron on the first layer is connected to every neuron on the second layer.
- **Partially connected**

  A neuron of the first layer does not have to be connected to all neurons on the second layer.
- **Feed forward**

  The neurons on the first layer send their output to the neurons on the second layer, but they do not receive any input back form the neurons on the second layer.
- **Bi-directional**

  There is another set of connections carrying the output of the neurons of the second layer into the neurons of the first layer.

Feed forward and bi-directional connections could be fully- or partially connected.

- **Hierarchical**

  If a neural network has a hierarchical structure, the neurons of a lower layer may only communicate with neurons on the next level of layer.
- **Resonance**

  The layers have bi-directional connections, and they can continue sending messages across the connections a number of times until a certain condition is achieved.

2.8.2.2 Intra-layer connections

In more complex structures the neurons communicate among themselves within a layer, this is known as intra-layer connections. There are two types of intra-layer connections.

- **Recurrent**
  the neurons within a layer are fully- or partially connected to one another. After these neurons receive input form another layer, they communicate their outputs with one another a number of times before they are allowed to send their outputs to another layer.

  Generally some conditions among the neurons of the layer should be achieved before they communicate their outputs to another layer.

- **On-center/off surround**
  A neuron within a layer has excitatory connections to itself and its immediate neighbors, and has inhibitory connections to other neurons. One can imagine this type of connection as a competitive gang of neurons. Each gang excites itself and its gang members and inhibits all members of other gangs.

After a few rounds of signal interchange, the neurons with an active output value will win, and is allowed to update its and its gang member's weights.

- (There are two types of connections between two neurons, excitatory or inhibitory. In the excitatory connection, the output of one neuron increases the action potential of the neuron to which it is connected. When the connection type between two neurons is inhibitory, then the output of the neuron sending a message would reduce the activity or action potential of the receiving neuron. One causes the summing mechanism of the next neuron to add while the other causes it to subtract. One excites while the other inhibits.

2.8.3 Learning

The brain basically learns from experience. Neural networks are sometimes called machine-learning algorithms, because changing of its connection weights (training) causes the network to learn the solution to a problem. The strength of connection between the neurons is stored as a weight-value for the specific connection. The system learns new knowledge by adjusting these connection weights. This method is proven highly successful in training of multilayered neural nets.

The network is not just given reinforcement for how it is doing on a task. Information about errors is also filtered back through the system and is used to adjust the connections between the layers, thus improving performance. A form of supervised learning.

The learning ability of a neural network is determined by its architecture and by the algorithmic method chosen for training. The training method usually consists of one of two schemes:

1. **Unsupervised learning**
   The hidden neurons must find a way to organize themselves without help from the outside. In this approach, no sample outputs are provided to the network against which it can measure its predictive performance for a given vector of inputs. This is learning by doing.
2. **Reinforcement learning**
   This method works on reinforcement from the outside. The connections among the neurons in the hidden layer are randomly arranged, then reshuffled as the network is told how close it is to solving the problem.

Reinforcement learning is also called supervised learning, because it requires a teacher. The teacher may be a training set of data or an observer who grades the performance of the network results.

Both unsupervised and reinforcement suffers from relative slowness and inefficiency relying on a random shuffling to find the proper connection weights.

2.8.3.1 Off-line or On-line

One can categorize the learning methods into yet another group, off-line or on-line. When the system uses input data to change its weights to learn the domain knowledge, the system could be in training mode or learning mode.

When the system is being used as a decision aid to make recommendations, it is in the operation mode; this is also sometimes called recall.

- **Off-line**
  In the off-line learning methods, once the systems enters into the operation mode, its weights are fixed and do not change any more. Most of the networks are of the off-line learning type.

- **On-line**
  In on-line or real time learning, when the system is in operating mode (recall), it continues to learn while being used as a decision tool. This type of learning has a more complex design structure.

2.8.3.2 Learning laws

There are a variety of learning laws, which are in common use. These laws are mathematical algorithms used to update the connection weights. Most of these laws are some sorts of variation of the best-known and oldest learning law, Hebb's Rule. Man's understanding of how neural processing actually works is very limited.

Learning is certainly more complex than the simplification represented by the learning laws currently developed. Research into different learning functions continues as new ideas routinely show up in trade publications etc. A few of the major laws are given as an example below.

- **Hebb's Rule**

  The first and the best-known learning Donald Hebb introduced rule. The organization of Behavior in 1949.This basic rule is: If a neuron receives an input from another neuron, and if both are highly active (mathematically have the same sign), the weight between the neurons should be strengthened.

- **Hopfield Law**

  This law is similar to Hebb's Rule with the exception that it specifies the magnitude of the strengthening or weakening.

It states, "if the desired output and the input are both active or both inactive, increment the connection weight by the learning rate, otherwise decrement the weight by the learning rate.

Most learning functions have some provision for a learning rate, or learning constant. Usually this term is positive and between zero and one.

- **The Delta Rule**

  The Delta Rule is a further variation of Hebb's Rule, and it is one of the most commonly used. This rule is based on the idea of continuously modifying the strengths of the input connections to reduce the difference (the delta) between the desired output value and the actual output of a neuron.

This rule changes the connection weights in the way that minimizes the mean squared error of the network. The error is back propagated into previous layers one layer at a time. The process of back-propagating the network errors continues until the first layer is reached. The network type called Feed forward, Back-propagation derives its name from this method of computing the error term.

This rule is also referred to as the Windrow-Hoff Learning Rule and the Least Mean Square Learning Rule.

- **Kohonen's Learning Law**

Learning inspired this procedure, developed by Teuvo Kohonen, in biological systems. In this procedure, the neurons compete for the opportunity to learn, or to update their weights. The processing neuron with the largest output is declared the winner and has the capability of inhibiting its competitors as well as exciting its neighbors. Only the winner is permitted output, and only the winner plus its neighbors are allowed to update their connection weights. The Kohonen rule does not require desired output. Therefore it is implemented in the unsupervised methods of learning.

Kohonen has used this rule combined with the on-center/off-surround intra- layer connection to create the self-organizing neural network, which has an unsupervised learning method. On this Internet site by Sue Becker you may see an interactive demonstration of a Kohonen network, which may give you a better understanding. [5]

2.9 How Neural Networks Learn?

Artificial neural networks are typically composed of interconnected "units", which serve as model neurons. The function of the *synapse* is modeled by a modifiable weight, which is associated with each connection. Each unit converts the pattern of incoming activities that it receives into a single outgoing activity that it broadcasts to other units. It performs this conversion in two stages:

1. It multiplies each incoming activity by the weight on the connection and adds together all these weighted inputs to get a quantity called the *total input.*
2. A unit uses an input-output function that transforms the total input into the outgoing activity.

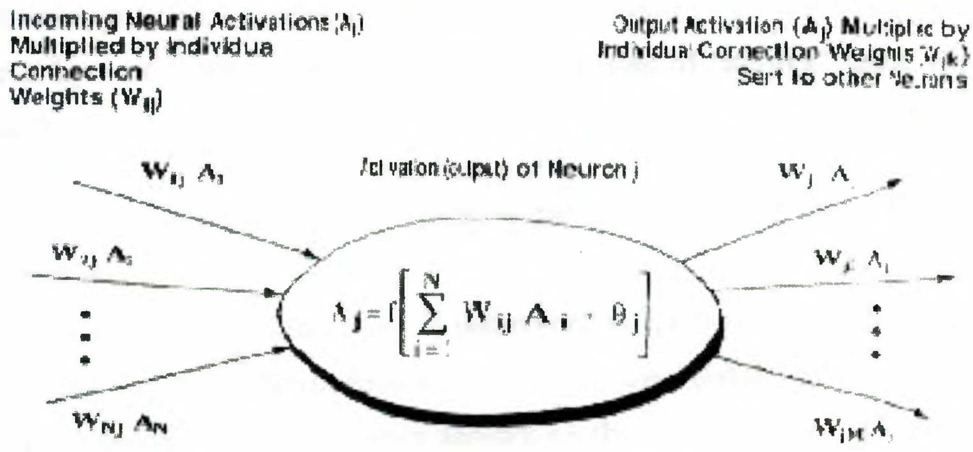


Figure 2.7. The weights and the input-output function (transfer function)

The behavior of an ANN (Artificial Neural Network) depends on both the weights and the input-output function (transfer function) that is specified for the units.

This function typically falls into one of three categories:

- Linear
- Threshold
- Sigmoid

For **linear units**, the output activity is proportional to the total weighted output.

For **threshold units**, the output is set at one of two levels, depending on whether the total input is greater than or less than some threshold value.

For **sigmoid units**, the output varies continuously but not linearly as the input changes. Sigmoid units bear a greater resemblance to real neurons than do linear or threshold units, but all three must be considered rough approximations.

To make a neural network that performs some specific task, we must choose how the units are connected to one another, and we must set the weights on the connections appropriately. The connections determine whether it is possible for one unit to influence another. The weights specify the strength of the influence.

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

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

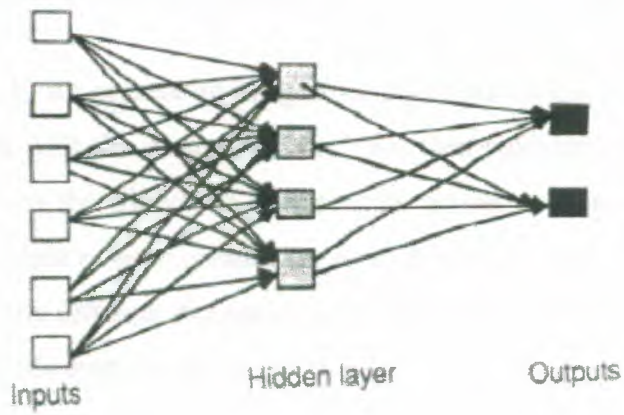


Figure 2.8. Simple type of network

This simple type of network is interesting because the hidden units are free to construct their own representations of the input.

The weights between the input and hidden units determine when each hidden unit is active, and so by modifying these weights, a hidden unit can choose what it represents.

We can teach a three-layer network to perform a particular task by using the following procedure:

1. We present the network with training examples, which consist of a pattern of activities for the input units together with the desired pattern of activities for the output units.
2. We determine how closely the actual output of the network matches the desired output.

3. We change the weight of each connection so that the network produces a better approximation of the desired output.

An Example to illustrate the above teaching procedure:

Assume that we want a network to recognize hand-written digits. We might use an array of, say, 256 sensors, each recording the presence or absence of ink in a small area of a single digit. The network would therefore need 256 input units (one for each sensor), 10 output units (one for each kind of digit) and a number of hidden units.

For each kind of digit recorded by the sensors, the network should produce high activity in the appropriate output unit and low activity in the other output units.

To train the network, we present an image of a digit and compare the actual activity of the 10 output units with the desired activity. We then calculate the error, which is defined as the square of the difference between the actual and the desired activities. Next we change the weight of each connection so as to reduce the error. We repeat this training process for many different images of each different images of each kind of digit until the network classifies every image correctly.

To implement this procedure we need to calculate the error derivative for the weight (EW) in order to change the weight by an amount that is proportional to the rate at which the error changes as the weight is changed. One way to calculate the EW is to perturb a weight slightly and observe how the error changes. But that method is inefficient because it requires a separate perturbation for each of the many weights. Another way to calculate the EW is to use the Back-propagation algorithm which is described below, and has become nowadays one of the most important tools for training neural networks.

2.10 Building A Neural Network

Since 1958, when psychologist Frank Rosenblatt proposed the "Perceptron," a pattern recognition device with learning capabilities, the hierarchical neural network has been the most widely studied form of network structure. A hierarchical neural network is one that links multiple neurons together hierarchically.

The special characteristic of this type of network is its simple dynamics. That is, when a signal is input into the input layer, it is propagated to the next layer by the interconnections between the neurons.

Simple processing is performed on this signal by the neurons of the receiving layer prior to its being propagated on to the next layer. This process is repeated until the signal reaches the output layer completing the processing process for that signal.

The manner in which the various neurons in the intermediary (hidden) layers process the input signal will determine the kind of output signal it becomes (how it is transformed).

As you can see, then, hierarchical network dynamics are determined by the weight and threshold parameters of each of their units. If input signals can be transformed to the proper output signals by adjusting these values (parameters), then hierarchical networks can be used effectively to perform information processing.

Since it is difficult to accurately determine multiple parameter values, a learning method is employed. This involves creating a network that randomly determines parameter values. This network is then used to carry out input-to-output transformations for actual problems. The correct final parameters are obtained by properly modifying the parameters in accordance with the errors that the network makes in the process. Quite a few such learning methods have been proposed. Probably the most representative of these is the error back-propagation learning method proposed by D. E. Rumelhart et al. in 1986. This learning method has played a major role in the recent neurocomputing boom.

The back-propagation paradigm has been tested in numerous applications including bond rating, mortgage application evaluation, protein structure determination, backgammon playing, and handwritten digit recognition. Choosing the right methodology, or back propagation algorithm, is another important consideration.

In working with the financial applications, many have found that the back-propagation algorithm can be very slow. Without using advanced learning techniques to speed the process up, it is hard to effectively apply back propagation to real-world problems.

Over fitting of a neural network model is another area, which can cause beginners difficulty. Over fitting happens when an ANN model is trained on one set of data, and it learns that data too well. This may cause the model to have poor generalization abilities - the model may instead give quite poor results for other sets of data.

For an in-depth coverage of other neural network models and their learning algorithms, please refer to the Technical Reading at the end of this User's Guide, the Technical Reference (sold separately), those papers listed in the Reference, or any other reference books on neural networks and relevant technology. [2]

2.11 A Learning Process

For decades there have been attempts to create computer programs that can *learn* like people - Artificial Intelligence. For example, how do you teach a child to recognize what a chair is? You show him examples telling him "This is a chair ; That one is not a chair" until the child learns the concept of what a chair is. In this stage, the child can look at the examples we have shown him and answer correctly to the question "Is this object a chair?". Furthermore, if we show to the child new objects, that he didn't see before, we could expect him to recognize correctly whether the new object is a chair or not, providing that we've given him enough positive and negative examples. This is exactly the idea behind the perceptron.

2.12 The Perceptron

The perceptron is a program that learns **concepts**, i.e. it can learn to respond with *True* (1) or *False* (0) for inputs we present to it, by repeatedly "studying" examples presented to it.

The Perceptron is a single layer neural network whose weights and biases could be trained to produce a correct target vector when presented with the corresponding input vector. The training technique used is called *the perceptron-learning rule*.

The perceptron generated great interest due to its ability to **generalize** from its training vectors and work with randomly distributed connections. Perceptrons are especially suited for simple problems in pattern classification. The perceptron looks like:

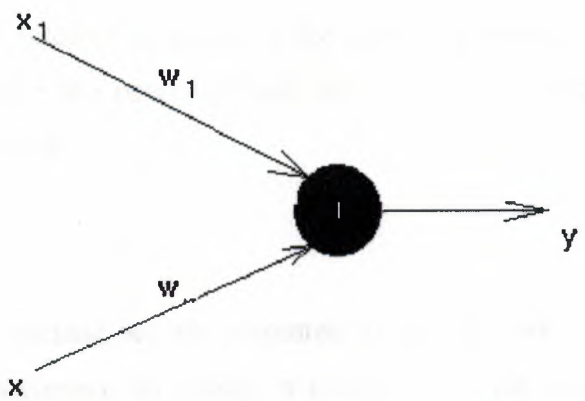


Figure 2.9. Single perceptron

Our perceptron network consists of a single neuron connected to two inputs through a set of 2 weights, with an additional bias input.

The perceptron calculates its output using the following equation:

$$P * W + b > 0$$

Where P is the input vector presented to the network, W is the vector of weights and b is the bias. [3]

2.12.1 The Learning Rule

The perceptron is trained to respond to each input vector with a corresponding target output of either 0 or 1. The learning rule has been proven to converge on a solution in finite time if a solution exists.

The learning rule can be summarized in the following two equations:

For all i:

$$W(i) = W(i) + [T - A] * P(i)$$

$$b = b + [T - A]$$

Where W is the vector of weights, P is the input vector presented to the network, T is the correct result that the neuron should have shown, A is the actual output of the neuron, and b is the bias.

2.12.2 Training

Vectors from a training set are presented to the network one after another. If the network's output is correct, no change is made. Otherwise, the weights and biases are updated using the perceptron-learning rule. An entire pass through all of the input training vectors is called an *epoch*. When such an entire pass of the training set has occurred without error, training is complete. At this time any input training vector may be presented to the network and it will respond with the correct output vector.

If a vector P not in the training set is presented to the network, the network will tend to exhibit *generalization* by responding with an output similar to target vectors for input vectors close to the previously unseen input vector P.

2.12.3 Limitations

Perceptron networks have several limitations. First, the output values of a perceptron can take on only one of two values (True or False). Second, perceptrons can only classify **linearly separable** sets of vectors. If a straight line or plane can be drawn to separate the input vectors into their correct categories, the input vectors are linearly separable and the perceptron will find the solution. If the vectors are not linearly separable learning will never reach a point where all vectors are classified properly.

The most famous example of the perceptron's inability to solve problems with linearly no separable vectors is the Boolean exclusive-or problem.

2.12.4 Developments from the simple perceptron:

Back-Propagated Delta Rule Networks (BP) (sometimes known and multi-layer perceptrons (MLPs)) and Radial Basis Function Networks (RBF) are both well-known developments of the Delta rule for single layer networks (itself a development of the Perceptron Learning Rule). Both can learn arbitrary mappings or classifications. Further, the inputs (and outputs) can have real values. [3]

2.12.5 Implementation

We implemented a single neuron perceptron with 2 inputs. The input for the neuron can be taken from a graphic user interface, by clicking on points in a board. A click with the left mouse button generates a '+' sign on the board, marking that it's a point where the perceptron should respond with 'True'. A click with the right mouse button generates a '-' sign on the board, marking that it's a point where the perceptron should respond with 'False'. When enough points have been entered, the user can click on 'Start', which will introduce these points as inputs to the perceptron, have it learn these input vectors and show a line which corresponds to the linear division of the plane into regions of opposite neuron response. [3]

2.13 A description of the Back Propagation Algorithm

To train a neural network to perform some task, we must adjust the weights of each unit in such a way that the error between the desired output and the actual output is reduced. This process requires that the neural network compute the error derivative of the weights (**EW**). In other words, it must calculate how the error changes as each weight is increased or decreased slightly. The back propagation algorithm is the most widely used method for determining the **EW**.The back-propagation algorithm is easiest to understand if all the units in the network are linear.

The algorithm computes each **EW** by first computing the **EA**, the rate at which the error changes as the activity level of a unit is changed. For output units, the **EA** is simply the difference between the actual and the desired output.

To compute the **EA** for a hidden unit in the layer just before the output layer, we first identify all the weights between that hidden unit and the output units to which it is connected. We then multiply those weights by the **EA**s of those output units and add the products. This sum equals the **EA** for the chosen hidden unit. After calculating all the **EA**s in the hidden layer just before the output layer, we can compute in like fashion the **EA**s for other layers, moving from layer to layer in a direction opposite to the way activities propagate through the network. This is what gives back propagation its name. Once the **EA** has been computed for a unit, it is straightforward to compute the **EW** for each incoming connection of the unit. The **EW** is the product of the EA and the activity through the incoming connection. Note that for non-linear units, the back-propagation algorithm includes an extra step. Before back propagating, the **EA** must be converted into the **EI**, the rate at which the error changes as the total input received by a unit is changed. [10]

A Back-Propagation Network Example

In this example a back-propagation network would be used to solve a specific problem, that one of an X-OR logic gate. That means that patterns have (1,1) should produce a value close to zero in the output node, and input patterns of (1,0) or (0,1) should produce a value near one in the output node. Finding a set of connection weights for this task is not easy; it requires application of the back-propagation algorithm for several thousand iterations to achieve a good set of connection weights and neuron thresholds.

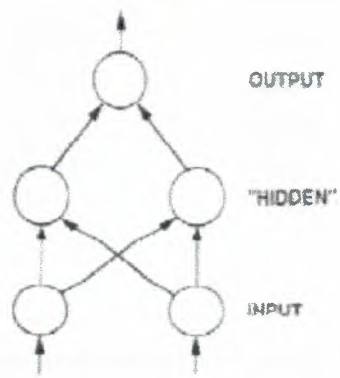


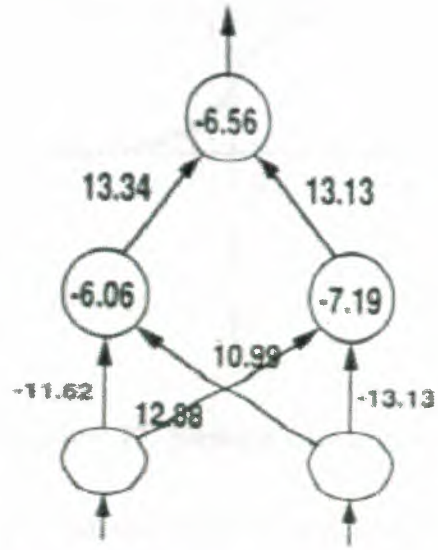
Figure 2.10. The basic architecture of neural network

The basic architecture for this problem has two input nodes, two hidden nodes, and a single output node as shown above.

This structure has variable thresholds on the two hidden and one output node (unit). This means that there are a total of 9 variables in the system:

- 4 weights connecting the input to the hidden nodes
- 2 weights connecting the hidden to the output node
- 3 thresholds.

Suppose we put in a pattern, say (0,1). That mean that there is 0 activation in the left-hand neuron on the first layer and an activation of 1 in the neuron on the right.



Results for Input Patterns:

INPUT	OUTPUT
(0, 0)	0.057
(1, 0)	0.949
(0, 1)	0.946
(1, 1)	0.052

Figure 2.11.The basic architecture of neural network with values

Now we move our attention to the next layer up. For each neuron in this layer, we calculate an input, which is the weighted sum of all the activations from the first layer.

The weighted sum is achieved by vector multiplying the activations in the first layer by a "connection matrix".

In our case we get a value of 0*(-11,62)+ 1*(10,99) = 10,99 for the neuron on the left in the second layer, and 0*(12,88) +1*(13,13) = =13,13 for the neuron on the right.

we add a "threshold" value (which is found for each neuron using the back-propagation rule), and apply an input-output (transfer) function.

The transfer function is defined for each different network. In our case it is a sigmoid:

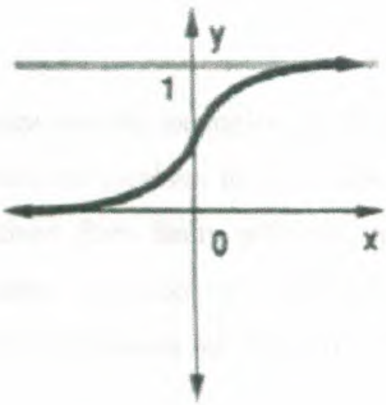


Figure 2.12. Sigmoid function curve

In this case it has been shown, that the activation of the neuron on the left side of the hidden (middle) layer is the transfer function applied to the difference (10,99-6,06) = 4,94. Applying the transfer function yields an activation value close to 1. The activation of the neuron on the right is the transfer function applied to (-13,13+7,19) = -5,14. Applying the transfer function yields a value close to 0. [10]

Approximating the next step, we use a value of 1 for the activation of the neuron on the left, and 0 for the neuron on the right, multiply each activation by its appropriate connection weight, and sum the values as input to the topmost neuron. This is approximately 1*(13,34)+0*(13,13) = 13,34. We add the threshold of -6,56 to obtain a value of 6,78. Applying the transfer function to it will yield a value close to 1 (0,946), which is the desired result. Using the other 3 binary input patterns, we can similarly show that this network yields the desired classification within an acceptable tolerance.

CHAPTER THREE
IMPLEMENTATIONS
OF
NEURAL NETWORKS

3.1 Overview

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

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

3.2 How Brain Maker Neural Networks work?

Neural networks are named after the cells in the human brain that perform intelligent operations. The brain is made up of billions of neuron cells. Each of these cells is like a tiny computer with extremely limited capabilities; however, connected together, these cells form the most intelligent system known. Neural networks are formed from hundreds or thousands of simulated neurons connected together in much the same way as the brain's neurons. Just like people, neural networks learn from experience, not from programming. Neural networks are good at pattern recognition, generalization, and trend prediction. They are fast, tolerant of imperfect data, and do not need formulas or rules. [5]

Neural networks are trained by repeatedly presenting examples to the network. Each example includes both inputs (information you would use to make a decision) and outputs (the resulting decision, prediction, or response). Your network tries to learn each of your examples in turn, calculating its output based on the inputs you provided.

If the network output doesn't match the target output, BrainMaker corrects the network by changing its internal connections. This trial-and-error process continues until the network reaches your specified level of accuracy. Once the network is trained and tested, you can give it new input information, and it will produce a prediction. Designing your neural network is largely a matter of identifying which data is input, and what you want to predict, assess, classify, or recognize.

3.3 Why is it useful?

Neural nets 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 net learns by being given a *training set*. Due to its complex, non-linear structure, the neural net can find relationships in data those humans as unable to do.

3.4 Why doesn't it work all the time?

Neural nets can only learn if the training set consists of good examples. The old saying of *'garbage in- garbage out'* is doubly true for neural nets. Great care should be taken to present decor related 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 net cannot learn from just a few examples.

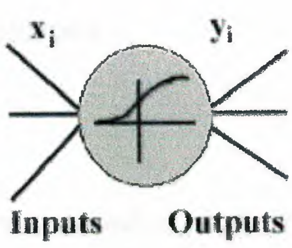


Figure 3.1.The neurons with input, output

3.5 What Applications Should Neural Networks Be Used For?

Neural networks are universal approximates, and they work best if the system you are using them to model has a high tolerance to error. One would therefore not be advised to use a neural network to balance one's checkbook! However they work very well for:

capturing associations or discovering regularities within a set of patterns; where the volume, number of variables or diversity of the data is very great; the relationships between variables are vaguely understood; or, the relationships are difficult to describe adequately with conventional approaches.

3.6 What Are Their Advantages Over Conventional Techniques?

Depending on the nature of the application and the strength of the internal data patterns you can generally expect a network to train quite well. This applies to problems where the relationships may be quite dynamic or non-linear.

ANNs provide an analytical alternative to conventional techniques, which are often limited by strict assumptions of normality, linearity, variable independence etc.

Because an ANN can capture many kinds of relationships it allows the user to quickly and relatively easily model phenomena which otherwise may have been very difficult or impossible to explain otherwise.

3.7 Neural Network Applications

3.7.1 Aerospace

- High performance aircraft autopilot, flight path simulation, aircraft control systems, autopilot enhancements, aircraft component simulation, aircraft component fault detection.

3.7.2 Automotive

- Automobile automatic guidance system, warranty activity analysis.

3.7.3 Banking

- Check and other document reading, credit application evaluation.

3.7.4 Credit Card Activity Checking

- Neural networks are used to spot unusual credit card activity that might possibly be associated with loss of a credit card.

3.7.5 Defense

Weapon steering, target tracking, object discrimination, facial recognition, new kinds of sensors, sonar, radar and image signal processing including data compression, feature extraction and noise suppression, signal/image identification.

3.7.6 Electronics

- Code sequence prediction, integrated circuit chip layout, process control, chip failure analysis, machine vision, voice synthesis, nonlinear modeling.

3.7.7 Entertainment

- Animation, special effects, market forecasting.

3.7.8 Financial

- Real estate appraisals, loan advisor, mortgage screening, corporate bond rating, credit-line use analysis, portfolio trading program, corporate financial analysis, and currency price prediction.

3.7.9 Industrial

- Neural networks are being trained to predict the output gasses of furnaces and other industrial processes. They then replace complex and costly equipment used for this purpose in the past.

3.7.10 Insurance

- Policy application evaluation, product optimization.

3.7.11 Manufacturing

- Manufacturing process control, product design and analysis, process and machine diagnosis, real-time particle identification, visual quality inspection systems, beer testing, welding quality analysis, paper quality prediction, computer-chip quality analysis, analysis of grinding operations, chemical product design analysis, machine maintenance analysis, project bidding, planning and management, dynamic modeling of chemical process system.

3.7.12 Medical

- Breast cancer cell analysis, EEG and ECG analysis, prosthesis design, optimization of transplant times, hospital expense reduction, hospital quality improvement, and emergency-room test advisement.

3.7.13 Oil and Gas

- Exploration

3.7.14 Robotics

- Trajectory control, forklift robot, manipulator controllers, vision systems

3.7.15 Speech

- Speech recognition, speech compression, vowel classification, text-to-speech synthesis.

3.7.16 Securities

- Market analysis, automatic bond rating, stock trading advisory systems.

3.7.17 Telecommunications

Image and data compression, automated information services, real-time translation of spoken language, customer payment processing systems.

3.7.18 Transportation

Truck brake diagnosis systems, vehicle scheduling, routing systems.[6]

3.8 Stocks, Commodities Applications

3.8.1 Neural Networks and Technical Analysis of Currencies

Mr. James O'Sullivan, of O'Sullivan Brothers Investments, Ltd. (Connecticut) has been successfully using many BrainMaker (California Scientific Software) neural networks on a daily basis for three years to do financial forecasting. [7]

He earned $250,000 in one month using neural networks to advise him on his New York Stock Exchange seat trades. Some of his networks are 88-90% accurate in their predictions, according to Mr. O'Sullivan. He uses an automated neural network system that monitors more than twenty different financial markets on a daily basis.

Mr. O'Sullivan has some unusual designs which act more as detectors of specific market conditions, rather than as exact price predictors. He combines the neural network data with other data from his technical analysis software to produce an automated report about a certain market. He gets his data live via satellite from Data Broadcasting and puts it into a charting and technical data module. He has pre-programmed the various algebraic manipulations to be performed on his data before BrainMaker files are built. He does moving averages, changes from period to period, and a few proprietary operations.

He runs new data through the system and produces the one-page report in about thirty seconds. He says at least 80% of his decision-making is based on neural network predictions.

Mr. O'Sullivan has not fully disclosed his neural network designs to us, but his basic insights are still quite valuable. The key is to ask the neural network the right kind of questions.

He asks questions such as "What is the probability of the product (or market) going up 0.618 standard deviations?" and "What percentage of the time does it go up that much?" He also asks questions about the directional behavior of the market and at what price the product (or market) is likely to take off in one direction or the other.

Mr. O'Sullivan's neural networks output several different things such as predicted prices, limits, and directional thrust. One neural network outputs the probability of a certain price occurring the next time period. Another neural network produces best stop price and best target price for long and short positions. Other neural networks produce directional indicators for three market energies. Another predicts the level at which the market is likely to take off.

In one design, the network is given various market conditions as input. The training output is the likelihood of various changes in price. For example, his neural network is told during training that, given similar market conditions, the closing price goes up at least 0.313 standard deviations above the prior day 90% of the time, at least 0.618 deviations 80% of the time, and at least 1 deviation 70% of the time.

An interesting phenomenon of the market is that when a change starts occurring in one direction or the other, there is a point at which it is very likely to continue moving in that direction for several time periods. Once a price reaches that level, there is a reduced risk to buy or sell (whichever is appropriate). Mr. O'Sullivan calls the network that predicts this price level his Risk Barometer network. He uses all the neural networks trained for a specific market when making decisions. For example, if the long-term trend is up, the Fast Movement network is a large positive number, his Risk Barometer says 233.092, and the NYSE is at 250, it could indicate an overreacted market that will reverse itself soon [17].

3.8.2 Predict Bond Prices with Neural Network software

G. R. Pugh & Company has been using a BrainMaker neural network trained on three-to-four years of historical data with an XT-compatible PC to help forecast the next year's corporate bond prices and ratings of 115 public utilities companies. "An XT is more than sufficient; it's a fast program," company president Mr. George Pugh notes.

Learning to use the program and create a neural network from scratch took only two days. The network trained itself to predict bond prices in about four hours.

G. R. Pugh & Company does consulting to predict bond prices for the public utility industry.

He maintains databases with financial and business information on the companies advises with business forecasts and credit risk assessments and predicts the financial and operating health of these companies. His expertise is also used by the brokerage industry. He advises clients by forecasting on the selection of good corporate bonds. His clients need to know more accurately which bonds represent good investments for their customers. Both increases and decreases in bond value provide the potential for profitable investment. Mr. Pugh announced that predicting bond prices with BrainMaker neural network software has been more successful than discriminate analysis and forecasting methods he has used, and even a little better than a person could do. "Discriminate analysis methods are good for getting the direction of lively issues, but neural networks pick up the subtle interactions much better," he explains. The network categorizes the ratings with 100% accuracy within a broad category and 95% accuracy within a subcategory. The mathematical method of discriminate analysis was only 85% accurate within a broad category. (Bonds are rated much like report cards, with broad category ratings such as A, B, C, etc. A subcategory could be A+, for example)." The network makes a significant contribution to his analysis. "The network allows me to pick up things that are not obvious with typical analysis," he says. Moreover, nearly all of the network's difficulties were found to be associated with companies that were experiencing a particularly unusual problem (such as regulatory risk) or had an atypical business relationship (such as being involved in a large sale and lease-back transaction). Ratings also tend to be subjective; financial items are not the only things considered by the rating companies. These influences were not represented in the training facts and this makes predictions difficult. The trained network forecasts next year's Standard & Poor's and Moody's corporate bond ratings (both are industry standards) from the previous year's S & P and Moody's ratings and 23 other measures of each company's financial strength, such as income, sales, returns on equity, five-year growth in sales, and measures of investment, construction, and debt load.

Each of these factors is assigned its own input neuron, and each company's ratings for next year are the outputs of the network. [29]

3.8.3 Predict the S&P 500 Index with Neural Network Software

A highly rated investment firm (Clearwater, FL) manages more than 60 million dollars in investments. LBS rely almost exclusively on computer techniques to guide its decisions in predicting the S&P 500 Index. The firm is a forerunner in using neural network software to recognize patterns and predict indexes and trends for financial forecasting.

The latest approach in forecasting used by LBS integrates an expert system with a BrainMaker neural network to make the most efficient use of the talents of each. The expert system provides rules, which govern the application of the neural network to the prediction. For example, if the expert system says the market is trending and the neural network forecasts the S&P will go up, then a buy signal is generated.

In predicting the S&P 500 index the neural network is trained with only recent market data (less than five years' worth) because it was found that the actual behavior of the market 25 years ago was not the same as it is today. Commonly available indicators are used such as the ADX, MACD, stochastic, DOW, volume, etc. The BrainMaker neural network "window" was found to be most effective at five market days for predicting the S&P 500 index. It was speculated that every weekday might have a certain "tone" to it, so that all Mondays tend to behave similarly. By presenting five-day intervals as historical input data while outputting forecasts for five days in advance, the neural network deals only with the same weekday for each prediction. [29]

The neural network trained by LBS predicts the S&P 500 with an average accuracy of 95%. This statistic was obtained by testing the network on hundreds of days it had never seen before. The network is retrained every night with the most current information to keep its behavior in agreement with the current behavior of the market [18].

3.8.4 Predicting Stock Prices using Neural Network Software

Warren Buffett is a pillar of the financial world, and with good reason. He has parlayed his theories on investing and market analysis into a substantial fortune, while others have used his advice to build their own highly successful investment portfolios. Some, too, have crunched Buffett's investment formulas, or something like them, into a suite of computer programs that produce an electronic version of the Buffett genius. [29]

Walkrich Investment Advisors, a consulting firm out of Cape Girardeau, Missouri, uses BrainMaker Neural Networks to do just that -- produce an investment tool (WRRAT) based loosely on Buffett's ideas and BrainMaker neural networks in predicting stock prices. How well does WRRAT perform in stock price prediction? From January '95 to January '96, a portfolio made up of WRRAT's 20 most under priced stocks would have seen an average advance of 32.63%, compared to the S&P's 31.93% gain over the same period. More recently, WRRAT's 20 most under priced stocks have enjoyed a 44.40% gain from January '96 to February '96, compared to the S&P's 38.65%.

How does WRRAT's forecasting compare to the flesh and blood Buffett? From July '95 to February '96, shares in Berkshire Hathaway, Buffett's holding company, have gained an average of 28%. With WRRAT's 1995 average advance of 32.63%, in financial forecasting Walkrich and BrainMaker can compete with the best.

Walkrich uses a BrainMaker neural network to determine the average premium (discount) the market is currently allocating to particular industries, and then uses that standard in an industry-by-industry neural network analysis designed to determine which stocks are trading below their market value. In that analysis, the neural network will appraise each stock, giving a price estimate (based on price/earnings, price/book and dividend yield), which is adjusted for size, industry, exchange listing and institutional influence.

The neural net's per-stock price estimate is then compared to the corresponding industry average, producing a calculated measure of each stock's relative value -- in short, whether the stock is being under priced or overpriced by the market.

3.8.5 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.

3.8.6 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.
- Computrac/Metastock, Telescan, and ASCII Data Formats Supported.

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."

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 detruding, 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.

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 [29].

3.9 Medical Applications

3.9.1 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, and 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. [34]

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.

3.9.2 Neural network Improves Hospital Treatment and 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.

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 numbers of diagnoses and physicians included in the program have 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 nonsurgical), 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 [34].

3.9.3 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.

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 [12].

3.9.4 Diagnose Heart Attacks with 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.

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. [34]

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.

3.9.5 Neural Network Orders Medical Laboratory Tests for ER

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 labs the neural network can order tests. 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.

Dr. Berkov is working on a neural network that is even more robust and focused. He is waiting for funds from the hospital to pursue this project on a larger scale and implement it in the hospital [12].

3.9.6 Classifying Psychiatric Patients for Care with Neural Networks

Dr. George Davis of the Augusta Mental Health Center (Augusta, ME) has trained a BrainMaker neural network which predicts the length of stay (LOS) for psychiatric patients. His system (available through Psybernetics, Inc. Augusta, ME) allows state hospitals and private facilities to determine which patients would benefit most from short stays and which require long-term (thirty days or more) hospitalization.

The system has the potential of providing an annual savings of $100,000 to a 300 bed private facility, and up to $600,000 for a tertiary (state) facility. Separating short term from long term stay patients upon admission rather than after some period of observation saves time and money.

Fewer inappropriate hospitalizations occur, which not only saves the state money, but allows the short term patient to benefit from community settings and support systems, and reduces the psycho-social stigma of hospitalization. In addition, there is a lessened burden on the legal system and law enforcement agencies, and less paperwork. Short-term patients who require hospitalization are more likely to be admitted to a general hospital because they may still retain insurance benefits.

The neural network performs better than traditional approaches in predicting the length of stay (LOS). Only 8-30% of the variance in LOS could be correlated to a combination of demographic, diagnostic and clinical variables. By comparison neural networks were able to explain 39-86% of the variance.

There are 48 or 49 kinds of input data used to train two different neural networks. The inputs include basic demographics, admission history, family support systems, ability to care for self, diagnosis information, etc.

There are four output neurons. The outputs define four classes of LOS: 1) less than 1 week, 2) greater than 1 week but less than thirty days, 3) greater than thirty days but less than six months, 4) greater than six months but less than one year.

Four networks were trained using 500-600 cases, two each on two different years of annual data. In this way, the predicted LOS for particular patients could be compared between two years at the institution, which had undergone major organizational changes.

Psychiatric diseases are the most difficult to predict. In addition, varying standards and funding policies make care more susceptible to chaos and difficult to compare between locations. Neural networks provide a means to predict the effectiveness of care for a specific location. Several neural networks can be trained with patient data from particular time periods, which will provide a method of judging the effectiveness of changing policy, procedure, or available resources.

For example, if several "typical" patient cases are created or particular troublesome cases are selected, these can be run through the different networks to determine which changes in treatment would be most beneficial [10].

3.9.7 Diagnosing Giant Cell Arthritis with Neural Network

Five doctors have trained a neural network using the American College of Hematology (ACR) database of patients with vacuities. The ACR has developed standards for classifying a number of rheumatic diseases. In addition to traditional classification approaches, other methods have been used such as decision trees, linear discriminate function analysis, logistic regression, and neural networks.

For the classification of Giant Cell Arthritis (GCA) of patients in the ACR database, all approaches have been used and compared. BrainMaker was trained on this set of patients, with the ACR diagnosis standards for comparison reasons. The inputs to the neural network were eight ACR predictor variables: 1) age greater than 50, 2) new localized headache, 3) temporal artery tenderness or decrease in a temporal artery pulse, 4) polymyalgia rheumatica, 5) abnormal artery biopsy, 6) erythrocyte sedimentation rate greater than 50mm/hour, 7) scalp tenderness or nodules, and 8) claudicating of the jaw, tongue or on swallowing. If the predictor variable was present, a 1 was input. If the variable was not present, a 0 was input. The output was a 1 if the patient was diagnosed as having GCA or a 0 if not.

There were 807 patients in the database, 214 with GCA and 593 with other forms of vacuities. The 807 patients were broken into three groups for neural network design and testing. One group of 80 or 81 patients was set aside for testing. A second group of 200 patients was set aside for monitoring the training (testing while training). A third group of 526 patients was set aside for training.

Ten sets of these triplet groups were created using a different set of 80 or 81 patients each time. Ten different neural networks were trained on slightly different training groups.

After training, each network was tested on its corresponding testing set. In this way, the networks would test each and every case in the database without having seen the case during training. The trained networks correctly classified 94.4% of the testing cases that had GCA and 91.9% of the cases that did not have GCA [10].

3.10 Sports Applications

Predicting Thoroughbreds Finish Time with Neural Networks

When Don Emmons' neural network picked the winning horse in 17 out of 22 thoroughbred races at Detroit Race Course, he was astounded that a $195 program (BrainMaker) running on a PC could do so well. "I am amazed at the ability of software to consistently include the winning horse in three of its picks," said Don.

Designing a neural network is largely a matter of defining the problem well in your own mind. The most difficult aspects are deciding what information you're going to use and gathering it.

There are several known methods of successfully predicting horserace winners with neural networks BrainMaker Professional provides a program, which automates the design of “competition”, networks such a horserace predictors. The program "Compete" designs, tests and runs a network based upon comparisons of all the competing items, comparing them two at a time. The one that wins the largest number of 2-item comparisons is the overall winner. Each item is rated for its overall likelihood of winning.

Another design approach, which can be used with standard BrainMaker, uses the full number of competing items as input, such as ten horses. There are two difficulties with this. First, not all races have the same number of horses running. Second, There will be a lot of inputs and outputs, making it more difficult to train.

Don Emmon's design approaches the problem by looking at each horse individually and predicting how well it might do in the race. A separate neural network is trained for each horse with past performance information. Then the neural networks are given current information and the ones with the best results are selected as likely winners. When Don selected three horses as the possible winner of the race, 77% of the time one of them was the winner.

Information may be gathered from the Racing Form or from a computerized service such as the Equine Line of Jockey Club Information Systems, Inc. in Lexington, Kentucky.

BrainMaker will read several different databases or a plain text file. Each horse has its own file and every past race is on a different line in the file. Don started out with four pieces of information: post position, the number of horses in the race, the horse's finish time and the track record. An input file might look like this:

```
        Postposition #horses finish time  track record

race#1        4           8       106.10       101.05

race#2        2           8       115.34       101.05

race#3        6           7       127.22       101.05

(etc.)
```

A little work was done with the numbers before the network was trained. The finish time was divided by the track record to provide a common denominator between all horseraces. Linear regressions were done on sets of three consecutive races in order to rate the recent performance as improving, staying the same or getting worse.

The final design included the postposition, the number of horses, the finish time as a percent of track record, and the change in recent performance as inputs. The output was the predicted finish time of the horse's next race expressed as a percentage of track records.

```
             Post position--> | one     |

          Number of horses--> | horse   | --->predicted time

               Finish time--> | neural  |

     Change in performance--> | network |
```

NOTES: Times are expressed as a percentage of track records. The change in performance is recalculated with a linear regression equation is used to produce a slope which describes the horse's recent performances as improving, not changing or getting worse. While the network is being trained, the inputs (the left portion of the diagram) represent the information from a past race. The output is what the horse did at its next race. At least eight races worth of information are needed for training.

The program presents the races to the network one a time, over and over, until the network learns what the horse has done in the past. Don typically let the training continue until the network output numbers, which were 95% accurate. The training can be done just before the race since it takes about five minutes to complete. After the network is trained, it can be used to predict an upcoming race Information about the horse's last race is input, and the output is a prediction of what the horse will do at this (the next) race [11].

In order to yield the best results from this network, a race selection system was created:

1. No races with maiden horses or allowance races are used.
2. No races with more than nine horses are used.
3. Every horse must have at least eight past performances
4. Best odds for profit are with races that have at least a 37% long shot to favorite ratio.

3.11 Science

3.11.1 Neural Network Predicts Detrimental Solar Effects

Dr. Henrik Lundstedt of Lund Observatory, Sweden, 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. [6]

The major causes 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.

The neural networks were trained with seven months of solar data from various US databases from CSSA, Stanford, CA; NOAA/SEL, Boulder, CO; and SacPeak/AFGL, NM. A period of data from June 6 - 21, 1990 was omitted from the training data and used for testing. During that period three major storms and one minor storm occurred. However, the traditional prediction method (NOAA/SEL) predicted no major storms and one minor storm, which was in fact one of the major storms. The neural network did much better. It predicted two of the three major storms and the minor storm, and predicted a minor storm for the third major storm.

3.11.2 Neural Network Analysis of Tran membrane-spanning Protein Helices

Dr. George Dombi of Wayne State University has developed neural networks, which generalize common themes found in peptides of 25 amino acid lengths. The sequences were sorted into two groups: Tran membrane or nontransmembrance type. 1751 training examples were used. As a result of training, bacteriohodopsin was examined to determine the position of it 7 Tran membrane helices. [6]

Using several training and testing experimental procedures, test results were obtained with up to 98% accuracy. A symbolic rather than numeric representation was used. For example, the alanine position 22 was represented by the input A22 being either on or off.

3.11.3 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.

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.

The training set used 403 samples, approximately 100 for each sex/species combination. The network was tested on 57 samples. The network identified the species and sex of every mosquito in the testing set correctly [13].

3.11.4 Neural Network Processing for Spectroscopy

Stellar Net 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. Stellar Net'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 Spectra Net application to process the spectral data and make the appropriate recognition in real time.

Operating on PC hardware and using a BrainMaker neural network as the processing engine, Spectra Net 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 Stellar Net customer is using Spectra Net's neural network capability to identify and assure proper hydration in recently harvested onions.

Spectra Net 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, Spectra Net 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.

To help automate the data preparation process, Stellar Net includes a SNAKE utility in all Spectra Net software packages that allows rapid spectral data configuration for training and testing neural networks [10].

3.11.5 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%. [10]

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.

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 is 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.

Although only two years of training and testing data were available, the results achieved to date are believed to be reliable and consistent enough to be used for forecasting guidance. Since this was the original goal of the project, the use of BrainMaker neural networks to predict local rainfall is now expanding to locations in other parts of the country [10].

3.11.6 Using a neural network to predict El Nino

Since January of 1991, a research team at the National Oceanic and Atmospheric Administration in Boulder, Colorado has been training a neural network to predict El Nino. According to head researcher, Dr. Vernon Derr, the purpose of the study was to determine if a neural network could accurately predict warm and cold events in the Pacific Ocean, and to compare the prediction capabilities of the neural network to other methods, particularly the Persistence method. According to Dr. Derr, the neural network did surprisingly well.

Researchers defined an El Nino or warm event as a departure of more than 1 standard deviation larger than the long term mean in various regions of the Eastern Pacific ocean. If the standard deviation was 1 standard deviation below the long term mean, it was a cold event. While the Persistence method is often used to make weather and climate predictions, it is unable to forecast change or predict the onset of a new situation. The neural network on the other hand was able to show a correlation between the prediction of El Nino and the actual occurrences of warm and cold events in the Pacific.

The neural net proved to be a useful device for predicting out to about six months, and depending on the input data, could possibly be useful to the fishing industry.

Researchers used input data found in the Comprehensive Ocean Atmospheric Data Set (COADS). A COAD is worldwide oceans data giving the sea surface on a monthly basis since 1884. Because warm events occur every five to seven years, and because each event is unique in terms of duration, onset and decay, the statistical character of each even is quite varied. As a result, an event is difficult to predict by any means.

For input, Dr. Derr's team primarily used ship's data from various part of the Pacific Ocean dating back to Mathew Fauntainmaury who was the original oceanographer in the Navy. It includes wind, air temperatures, surface temperatures and southern oscillations, which is a comparison of sea surface and pressure between Darwin, Australia and Tahiti. It is a know fact that this difference in pressure occurs during El Ninos, but not before, so it is therefore not useful in predicting them.

One part of the research study was to determine the best set of data. According to Dr. Derr, "the set of data we used to predict things over the last year is probably not the ideal set, and we will be using a different set in the future." Because the team used most of the available data for training, only 10% of the data was left for validation and this remaining 10% may not even encompass a period in which El Nino occurred. The network was trained using the standard sigmoid transfer function. Using the genetic algorithm method, the team studied learning rates and tolerances to determine the best set for the data set they were using. They also varied the number of hidden neurons to determine the optimum number, but have not yet gone to more than 1 layer; although according to Dr. Derr that is something they want to do in the future.

Testing went as follows: In January of 1991, the team started predicting skill scores -- actually the RMS differences between the actual ocean temperature and the predicted temperature-- for up to six months ahead Then in February of 1991, they again predicted (on the basis of current data) for 1 to 6 months ahead. They continued in this manner up until the present time. According to Dr. Derr, "Those were quite good in the sense that the RMS skill differences were in the order of less than a degree averaged over a long period."

However one of the problems was that neither the Persistence method nor the neural net did a thorough job of predicting the onset of the warm or cold event. This fact leads Dr. Derr to speculate that the data was not sufficient for the purpose and that it should include not only at the sea surface temperature in Region 4 of the Pacific, but also at least a nine or ten year the history of it.

Dr. Derr plans on concluding his studies at the end of 1993. In the means time he plan to employ a rather unusual validation process. The team will train using all the data they have and then they're going to find skill scores for the same period of time again using all the data they have [6].

3.11.7 Using a Neural Network to Measure Air Quality

Researchers Eugene Yee and Jim Ho at the Defense Research Establishment Suffield, Chemical & Biological Defense Section, in Alberta, Canada have trained a neural network to recognize, classify and characterize aerosols of unknown origin with a high degree of accuracy. Their results hold considerable promise for applications to rapid real-time air monitoring in the areas of occupational health and air pollution standards.

Their research applied a neural network to the recognition and classification of environmental, bacterial, and artificial aerosols on the basis of the aerodynamic particle size distribution. Because of their variability, aerosols are difficult to recognize using conventional pattern recognition techniques. However, the health effects posed by airborne industrial, bacterial, and viral particles depend critically on the ability to recognize, characterize and classify these particles on the basis of their particle size distribution functions.

The input data was constructed from aerodynamic particle size distribution functions (PSDF) obtained from 11 different aerosol populations. The PSDF's were measured with an aerodynamic particle size, which determines the aerodynamic diameter of individual aerosol particles, by measuring the transit time of the particles between two spots generated by a laser velocimeter that employs a polarized laser light source. Size distributions were classified into 11 categories depending on the source of the aerosol particles generating the distribution.

It was found that a recognition rate of 100% was obtained for the training set using neural networks with three or more hidden units and that there was a smaller number of passes through the training data with an increase in the number of hidden units in the network. There was virtually no increase in the learning times of the networks with more than 10 hidden neurons. In addition, the performance of the networks did not deteriorate when the number of hidden units was increased beyond 10.

Experiments were also conducted to study the performance characteristics of the neural network as a function of the quality of data used for the training set and the test set and of the inclusion of random noise in the connection strengths of the trained network. Results showed that the neural network was more suited than conventional methods for classification of signals from systems where one is confronted with ignorance of the statistical characteristics of the noise corrupting the signals [11].

3.11.8 the use of Neural Networks in testing plastic quality

Monsanto is using BrainMaker to predict the quality of plastics to be used in windshields. The lab tests for brittleness and elasticity take up to 4 hours. Through use of a neural network they were able to save research time, processing time, and money.

Comments on the program:
"Very easy to use" -Edwin Nazarian
"Very good. Terrific manuals. My main reason for referring product to others" -Azmi Jaferey

Their network contained 18 different inputs including:
temperature,
length of time baked,
processing teqniques,
source of the chemicals, the network was able to predict the plastic quality within a 10% tolerance [11].

3.12 Manufacturing

3.12.1 The use of neural networks in testing plastic quality

Monsanto is using BrainMaker to predict the quality of plastics to be used in windshields. The lab tests for brittleness and elasticity take up to 4 hours. Through use of a neural network they were able to save research time, processing time, and money.

Comments on the program:
"Very easy to use" -Edwin Nazarian
"Very good. Terrific manuals. My main reason for referring product to others" -Azmi Jaferey

Their network contained 18 different inputs including:
temperature,
length of time baked,
processing teqniques,
source of the chemicals,
the network was able to predict the plastic quality within a 10% tolerance. [24]

3.12.2 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 Saigon, 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).

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.

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.

3.12.3 Neural Network performs 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.

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. (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.)

3.12.4 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.

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 glycoside, reaction temperature, pressure, amount of enzyme, and stirring speed were varied. The synthesis yielded ethyl 6-O-dodecanoyl D-glucopyranoside.

This experimental data was used to train the neural network to output the amount of the 6-O monoester and a diester by-product, represented as a percentage of yields.

The neural network had three layers: 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 diester. One of these cases was tested in the lab and the results were close to experimental observation. The network had predicted 88.1% monoester and the experiment yielded 86.2%. The network predicted 4.0% diester, the experiment yielded 4.8%. [24]

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.

3.12.5 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. [24]

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.

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.

3.13 Pattern Recognition

3.13.1 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, Octel Communications, the world's largest provider of voice and fax information processing and services, acquired Compass Technology. 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 normalize (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. [33]

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.

3.13.2 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. 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 BrainMaker running on the BrainMaker 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.(1) 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.

3.13.3 Chaos, Strange Attractors and Neural Network 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.

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. Some that believes it because there is an underlying mechanism analytic approach 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 no linearity and feedback. If you discover the underlying math that explains this, please call us immediately.

3.13.4 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. (1) 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.(2) 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. [33]

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.

3.13.5 Decoding Algorithms and Predicting Sequences with Neural Networks

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 BrainMaker 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. [33]

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.

3.14 Conclusion

The computing world has a lot to gain fron neural networks. Their ability to learn by example makes them very flexible and powerful. Furthermore there is no need to devise an algorithm in order to perform a specific task; i.e. there is no need to understand the internal mechanisms of that task. They are also very well suited for real time systems because of their fast responseand computational times which are due to their parallel architecture.

Neural networks also contribute to other areas of research such as neurology and psychology. They are regularly used to model parts of living organisms and to investigate the internal mechanisms of the brain.

Perhaps the most exciting aspect of neural networks is the possibility that some day 'consious' networks might be produced. There is a number of scientists arguing that conciousness is a 'mechanical' property and that 'consious' neural networks are a realistic possibility.

Finally, I would like to state that even though neural networks have a huge potential we will only get the best of them when they are intergrated with computing, AI, fuzzy logic and related subjects.

CHAPTER FOUR

NEURAL NETWORKS IN BUSINESS, MANAGEMENT AND FINANCE

4.1 Overview

There are different types of neural network models which are applicable when solving business problems. The history of neural networks in business is outlined, leading to a discussion of the current applications in business including data mining, as well as the current research directions. The role of neural networks as a modern operations research tool is discussed. Scope and purpose Neural networks are becoming increasingly popular in business. Many organisations are investing in neural network and data mining solutions to problems which have traditionally fallen under the responsibility of operations research. Business is a diverted field with several general areas of specialisation such as accounting or financial analysis.

Almost any neural network application would fit into one business area or financial analysis. There is some potential for using neural networks for business purposes, including resource allocation and scheduling. There is also a strong potential for using neural networks for database mining, that is, searching for patterns implicit within the explicitly stored information in databases. Most of the funded work in this area is classified as proprietary. Thus, it is not possible to report on the full extent of the work going on. Most work is applying neural networks, such as the Hopfield-Tank network for optimization and scheduling.

4.2 The Beginning of Neural Networks in the business

The 1988 DARPA Neural Network Study [DARP88] lists various neural network applications, beginning in about 1984 with the adaptive channel equalizer. This device, which is an outstanding commercial success, is a single- neuron network used in long-distance telephone systems to stabilize voice signals. The *DARPA* report goes on to list other commercial applications, including a small word recognizer, a process monitor, a sonar classifier, and a risk analysis system. Neural networks have been applied in many other fields since the *DARPA* report was written. A list of some applications mentioned in the literature follows. [8]

4.3 Neural Network Predicts Gas Index Prices

Dr. Al Behrens of Northern Natural Gas in Nebraska has developed a neural network that predicts next month's gas price change with an average accuracy of 97%. Northern Natural Gas is a regulated wholesaler of natural gas. They must develop and file a rate for gas based on the volume-weighted average cost of gas.

Prices and terms are specified in contracts. Being able to predict costs provides a valuable piece of planning data.

The monthly price is sometimes tied to an index such as those published in Inside FERC and Natural Gas Week. The price is a function of many factors, including recent market activity, seasonal factors, weather, etc. Dr. Behrens used seven inputs to the neural network, which included some past information. [25]

4.4 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!" [28]

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 known facts 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."

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.

4.5 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."

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 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. [9]

4.6 Real Estate Appraisal with Neural Networks

Neural networks can be used to predict the sale price of a home. The information provided by the neural network helps appraisers make assessments, helps sellers determine appropriate asking prices, and helps homeowners decide if improvements would be cost-effective. As the neural network designer, your primary responsibilities are to clearly define the problem and present the data in such a way that the network can find patterns. Once this is accomplished, training the network is mostly a fine-tuning process.

The Problem

Traditional methods for determining the value of real estate include appraisal by a certified expert, computer-assisted appraisal and, of course, the actual sale price. The problems inherent with these valuation methods are the inconsistency between appraisers, the inability of machines to consider more than rules and mathematical formulas, and the effects of changing market conditions.

Neural networks do not fall victim to these problems. When applied to real estate appraisal, neural networks are able to predict the actual sale price of properties with 90% accuracy. Neural networks perform better than multi-variety analysis, since networks are inherently nonlinear. They can also evaluate subjective information, such as a neighborhood rating, which is difficult to incorporate into traditional mathematical approaches. Richard Burst, a Senior Vice President at Day & Zimmerman, Inc., the nation's leading provider of mass appraisal services to state and local governments, has successfully trained a neural network to appraise real estate in the New York area. His network incorporates eighteen data items, which include the number of dwelling units, fireplaces, plumbing fixtures, square feet of living area, age, months since last sale, and air conditioning. He uses 217 sales records from 1988 and 1989 with prices ranging from $103,000 to $282,000. His network was trained on a 386 using BrainMaker Professional v2.5 (California Scientific Software: Nevada City, CA).

The Data

The data used in Mr. Borst's network, collected by the mass appraisal firm, Cole-Layer-Tremble, represent sales from a single area. The data chosen are similar to what an appraiser would examine to make an assessment. The table below lists all variables used in the original network design and the range of possible values for each. All values are continuous except two, heating type and neighborhood group. These two inputs represent categories, but since each has only two possible values, they don't need to be divided into separate inputs. [7]

4.7 Neural Network Red-Flags Police Officers With Potential For Misconduct

The Chicago Police Department has used BrainMaker to forecast which officers on the force are potential candidates for misbehavior. The Department's Internal Affairs Division used neural networks to study 200 officers who had been terminated for disciplinary reasons and developed a database of pattern-like characteristics, behaviors, and demographist found among the 200 police officers.

BrainMaker then compared current Department officers against the pattern gleaned from the 200-member control group and produced a list of officers who, by virtue of matching the pattern or sharing questionable characteristics to some degree, were deemed to be "at risk."

This particular application has been highly controversial, drawing criticism from several quarters - the most vocal being Chicago's Fraternal Order of Police. William Nolan, the Order's president, has made Orwellian references, saying the Department's program seems like "Big Brother." Scientific American, Playboy, New Scientist, and Law Enforcement News have all done articles on the ethical implications of the Chicago P.D.'s program with mixed reviews.

The C.P.D. Internal Affairs Division, however, was pleased with the results. After BrainMaker studied the records of the 12,500 current officers (records that included such information as age, education, sex, race, number of traffic accidents, reports of lost weapons or badges, marital status, performance reports, and frequency of sick leaves) the neural network produced a list of 91 at-risk men and women.

Of those 91 people, nearly half were found to be already enrolled in a counseling program founded by the personnel department to help officers guilty of misconduct. The I.A.D. now intends to make the neural network a supplement to the counseling program because, as Deputy Superintendent Raymond Risely said, the sheer size of the Chicago police force makes it "pretty much impossible for all at-risk individuals to be identified [by supervisors]."

Terry Heckart, a graduate student at Ohio's Bowling Green State University, recommended neural networks to Chicago's Internal Affairs Division. Heckhart told the Division officials that the software could be effective for two reasons: one, as the number of variables increase in the application, the output reliability increases; secondly, neural nets can deal with missing data. That, says Risley, "was really the key to solidifying our interest."

"We're very pleased with the outcome," Risley says, "We consider it much more efficient and capable of identifying at-risk personnel sooner than command officers might be able to do. The old method just can't compete with it."

The Chicago Police Department stresses that the program utilizing BrainMaker has no punitive ramifications. Risley notes that "it's not disciplinary . . . it's an opportunity for an officer who is moving in the wrong direction to rehabilitate himself . . . if an officer refuses to participate, nothing happens to him."

Despite the ethical discussion raging over whether a neural network should be used to monitor human beings, the program cannot be accused of being subjective and personally biased as "manned" programs often are. Clearly, the software can hold no personal grudges and seeks only to dispassionately identify patterns and characteristics that could spell trouble. The alternative system, being human based, cannot avoid subjectivity and bias on some level. It is worthy of note that the Fraternal Order of Police "vehemently opposed" the Department's old system for that very reason.

To counterbalance the inherent "dispassion" of the neural network, the Department closely examined the net's findings to ensure that officers who are clear anomalies, and thus don't warrant being on the list, are removed from consideration.

This combination of objective technology and subjective humanity does not necessarily spell perfection, but it does signify a promising move in that direction.

Currently, we are told, the Chicago Police Department does not use BrainMaker to forecast problems with officers. The program was apparently terminated due to its controversial nature. [9]

4.8 Managing Jury Summoning with Neural Network

The Intelligent Summoner from MEA (Norristown, PA) allows a courthouse to dramatically reduce the number of jurors called for potential service, saving an average of 25-40% of the cost of jurors. Courts often waste money by calling (and paying for) more potential jurors to show up and wait around than are needed for the cases being heard. The Montgomery Court House in Norristown, PA saves $70 million annually using this system.

The Intelligent Summoner determines the number of jurors needed for the next day at a specific courthouse. The program is custom-tailored for a specific courthouse by MEA with information about that courthouse.

At least one year's worth of past information about the courthouse is needed for custom tailoring. The information can be entered into the Intelligent Summoner system and includes dates, judges, types of cases, and number of jurors used at a particular courthouse. A file is written to disk, which is then sent to MEA.

MEA then creates and trains a neural network using BrainMaker, which learns the needs of that courthouse. The trained neural network is sent back to the courthouse and the file is read onto the PC. From then on, the system is ready for use at that courthouse by anyone with typing skills.

To use the system, a daily survey of tomorrow's judges, trial types, and size of jury panel is input with the data-entry portion of the program. The program will immediately provide the total number of jurors that should be called for tomorrow. [8]

4.9 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 BrainMaker 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, its an ideal application for BrainMaker.

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 BrainMaker 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.

While Professor Hannas 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 of historical record of work that has been done in particular sections of road over the last few years as well as an annual evaluation of the Riding Comfort Index.

The Riding Comfort Index is a rating of how comfortable you are on a particular section of the highway. The smoothest, best road would score 10 out of 10. A bumpy road for example would score a 5. Roads are measured before treatment and again a year later. If a year after treatment the road is still scores high it means the treatment was a good one. If the score is low it means the treatment wasn't really appropriate.

To validate his results, Professor Hanna is seeking funding to test his program. His testing will include more highway locations, different types of cracks and bumps, and data collected at different times of the year. He plans to record various sections of a particular highway; classify the cracks found there (severe, medium or light); look at external factors (annual daily traffic, air temperature, amount of snow removal, wind factor); input all facts into the software, and obtain recommendation for the ideal treatment.

BrainMaker transfer function will be used to determine the exact confidence -- based on information provided in the training phase. [25]

4.10 Applying Neural Networks to Predict Corporate Bankruptcy

This research is concerned with the investigation of the application of neural networks to the prediction of corporate bankruptcy. In the past, corporate failure prediction has been based on traditional methods of financial ratios analysis with multivariate discriminant analysis (MDA).

MDA has been sharply criticised because the validity of its results hinges on restrictive assumptions. Two assumptions are particularly problematic for ratio analysis. First, MDA requires that the decision set used to distinguish between distressed and viable firms must be linearly separable. Second, MDA does not allow for a ratio's signal to vacillate depending on its relationship to another ratio or set of ratios. The analysis of corporate evaluation is not an exact science and the complex nature of this domain makes a simple model of MDA and financial ratios analysis unsuitable for failure prediction.

This research seeks to investigate whether Artificial Neural Networks (ANN) can successfully discern patterns or trends in financial data and use them as early warning signals of distressful conditions in current viable firms. Neural networks are an information processing system which mimic human reasoning and have been used in various applications of financial modelling.

A critical review of the literature on Neural Networks in Financial Services has been completed. This review has shown that neural network approaches to date have demonstrated limited success. Previous research has failed to exploit domain knowledge and has utilised Artificial Neural Networks that are too simplistic. The present research proposes a ANN architecture based on linked networks built utilising domain knowledge.

Future work will be concerned with choosing a combination of individual networks to solve the problem in this domain with optimal generalisation performance. There are a number of issues to be tackled including the unpredictable interactions of the numerous design parameters of the networks, the boundaries of business environments, the complex nature of financial indicators, as well as the particular characteristics of the data involved.

This project will select three samples of company information from company's annual reports (e.g. the balance sheets, the cash flow statements, the profit and loss and income statements) and determine which variables are relevant as indicators of financial failure by utilising domain knowledge. The variables selected for each sample company information will be used as inputs to individual networks. Each architecture will be defined by the arrangement of its nodes, that is, the set of all weighted connections. Various feed forward networks will be considered. The main aim is the development of a full methodology which will include inter-connected networks and link them together to produce optimal generalisation. The evaluation will use two years financial reports as exemplars for test purposes. [30]

4.11 Marketing

There is a marketing application which has been integrated with a neural network system. The Airline Marketing Tactician (a trademark abbreviated as AMT) is a computer system made of various intelligent technologies including expert systems. A feedforward neural network is integrated with the AMT and was trained using back-propagation to assist the marketing control of airline seat allocations. The adaptive neural approach was amenable to rule expression. Additionaly, the application's environment changed rapidly and constantly, which required a continuously adaptive solution. The system is used to monitor and recommend booking advice for each departure. Such information has a direct impact on the profitability of an airline and can provide a technological advantage for users of the system.

While it is significant that neural networks have been applied to this problem, it is also important to see that this intelligent technology can be integrated with expert systems and other approaches to make a functional system. Neural networks were used to discover the influence of undefined interactions by the various variables. While these interactions were not defined, they were used by the neural system to develop useful conclusions. It is also noteworthy to see that neural networks can influence the bottom line. [9]

4.12 Credit Evaluation

The HNC company, founded by Robert Hecht-Nielsen, has developed several neural network applications. One of them is the Credit Scoring system which increase the profitability of the existing model up to 27%. The HNC neural systems were also applied to mortgage screening. A neural network automated mortgage insurance underwritting system was developed by the Nestor Company. This system was trained with 5048 applications of which 2597 were certified. The data related to property and borrower qualifications. In a conservative mode the system agreed on the underwritters on 97% of the cases. In the liberal model the system agreed 84% of the cases. This is system run on an Apollo DN3000 and used 250K memory while processing a case file in approximately 1 sec. [7]

4.13 Technology of Neural Networks in the Business Environment

Today's businesses are using new technologies to address old problems. Internal auditors have an opportunity to do the same with the use of neural networks.Business strategist Daniel Burrus, author of Technotrends: How to Use Technology to Go Beyond Your Competition, asserts that to remain competitive, businesses must not only use the tools of tomorrow, but utilize them in new ways. Neural network technology equips people with expertise that previously could only have been attained with years of training and experience. Productivity and accuracy has also increased with the use of neural network based expert systems.

The applicability of the backpropagation architecture, combined with the rapid advance of computers and microprocessors, makes this technology feasible and more cost effective. Neural network techniques applied to the business environment - specifically in bond rating and stock price predictions - have outperformed widely used regression and discriminant analysis techniques. [29]

Neural networks are being used in a variety of applications. Some successful commercial applications include:

¨ American Express, Mellon Bank, and First USA Bank are studying patterns of credit card use and detecting questionable transactions.

¨ Merrill Lynch & Co., Salomon Brothers Inc., Citibank, and the World Bank are forecasting financial information.

¨ Gerber Baby Foods is managing cattle futures trade.

¨ Chase Manhattan Bank is evaluating corporate loan risk.

¨ Veratex Corporation and Spiegel are targeting potential catalog recipients and customers.

¨ Texaco is developing geological applications.

Conclusion

Neural networks represent a new technology with many potential uses. Their capabilities have already been proven in a variety of business applications.

Commercial software packages available make neural networks accessible for use by individuals familiar with spreadsheet or relational database software.

Now the challenge remains to discover new and creative ways of using neural networks in an internal auditing environment.

CONCLUSION

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

In addition, the future of the neural networks is wide open and may lead to many answers or questions. Because it is technology gives a computer system an amazing capacity to actually learn from input data. Artificial neural networks have provided solutions to problems normally requiring human observation and thought processes. Either humans or other computer techniques can use neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, to extract patterns and detect trends that are too complex to be noticed. And also the Neural Networks have many advantages such as Adaptive learning, Self-Organization, Real Time Operation, and Fault Tolerance via Redundant Information Coding...etc.

In chapter two the architectures of neural networks were described Networks such as the one just described are called artificial neural networks (ANNs), in the sense that they represent simplified models of natural nerve or neural networks. The basic processing element in the nervous system is the neuron. Neural networks process information in a similar way the human brain does. The network is composed of a large number of highly interconnected processing elements (neurons) working in parallel to solve a specific problem. The commonest type of artificial neural network consists of three groups, or layers, of units: a layer of "input" units is connected to a layer of "hidden" units, which is connected to a layer of "output" units.

Obviously the idea of designing the neural networks referred to the human brain behaviors. Therefore neural networks learn by example. They cannot be programmed to perform a specific task. The examples must be selected carefully otherwise useful time is wasted or even worse the network might be functioning incorrectly. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze.

Some neural network structures are not closely to the brain and some does not have a biological counterpart in the brain. However, neural networks have a strong similarity to the biological brain and therefore a great deal of the terminology is borrowed from neuroscience.

Neural networks are sometimes called machine-learning algorithms, because changing of its connection weights (training) causes the network to learn the solution to a problem. The strength of connection between the neurons is stored as a weight-value for the specific connection. The system learns new knowledge by adjusting these connection weights. This method is proven highly successful in training of multilayered neural nets.

The learning ability of a neural network is determined by its architecture and by the algorithmic method chosen for training. The training method usually consists of one of two schemes: Unsupervised learning, supervised learning. There are a variety of learning laws, which are in common use. These laws are mathematical algorithms used to update the connection weights.

Chapter three was aimed to show the fields, where the neural networks can be applied. So many applications of the neural network either in the real world or predicted applications., also used in the following specific paradigms: recognition of speakers in communications; diagnosis of hepatitis; recovery of telecommunications from faulty software; interpretation of multimeaning Chinese words; undersea mine detection; texture analysis; three-dimensional object recognition; handwritten word recognition; and facial recognition.

Neural networks are trained by repeatedly presenting examples to the network. Each example includes both inputs (information you would use to make a decision) and outputs (the resulting decision, prediction, or response). Your network tries to learn each of your examples in turn, calculating its output based on the inputs you provided. Neural networks are good at pattern recognition, generalization, and trend prediction. They are fast, tolerant of imperfect data, and do not need formulas or rules.

Neural networks work very well for: capturing associations or discovering regularities within a set of patterns; where the volume, number of variables or diversity of the data is very great; the relationships between variables are vaguely understood; or, the relationships are difficult to describe adequately with conventional approaches.

In additions, neural networks have implemented successfully in the practice such as Manufacturing, industries, science, busincess, medical applications...etc, therefore the neural networks are powerful, recent technology, wide open future and flexible techniques for solving the problems.

In chapter four, fields were presented an application of the neural networks in business, management and finance. There are different types of the neural networks models, which are applicable when solving business problems. Today's businesses are using new technologies to address old problems. The firms and companies that use the neural networks in their business and managements obtain a lot of benefits and interests because the neural networks have good predictions for optimizing the real business world especially in the marketing.

Finally Neural networks represent a new technology with many potential uses. Their capabilities have already been proven in a variety of business applications.

REFERENCES

[1] J. Anderson, "A Simple Neural Network Generating an Interactive Memory," Mathematical Biosciences 14:197-220, 1972.

[2] M.H. Hassoun and D.W. Clark, "An Adaptive Attentive Learning Algorithm for Single Layer Neural Networks," in Learning Algorithms II, Proceedings of the First Annual IEEE Conference on Neural Networks, vol. I, pp. 431-440, 1988.

[3] F. Rosenblatt, Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms, Spartan Books, Washington, D.C., 1962.

[4] Eberhart RC and Dobbins RW (1990) Early neural network development history: The age of Camelot. IEEE Engineering in Medicine and Biology 9, 15-18.

[5] Hinton G.E. (1992) how neural networks learn from experience. Scientific American 267, 144-151.

[6] Swingler K. (1996) applying neural networks. A Practical Guide. Pub Academic Press, NY. ISBN 0-12-679170-8.

[7] Wong, Bo K. Bodnovich, Thomas A. Selvi, Yakup. (1995) Bibliography of neural network business applications research: 1988-September 1994. Expert Systems. v 12. p 253-262.

[8] Kaastra I. Boyd M. (1996) designing a neural network for forecasting financial and economic time series. Neurocomputing. 10, 215-236 Apr.

[9] Poddig T. Rehkugler H (1996) A world model of integrated financial markets using artificial neural networks. Neurocomputing. 10, 251-273.

[10] Astion ML, Wilding PW (1992) The application of backpropagation neural networks to problems in pathology and laboratory medicine. Arch Pathol Lab Med 116: 995-1001.

[11] Maddalena DJ. (1996) Applications of artificial neural networks in quantitative structure activity relationships. Exp Opin Ther Patents 6, 239-251.

[12] Baxt WG. (1995) Application of artificial neural networks to clinical medicine. [Review] Lancet. 346 (8983): 1135-8, Oct 28. .

[13] Neural Networks (1993) by Eric Davalo and Patrick Naim

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

[15] An introduction to neural computing (1991). Aleksander, I. and Morton, H. 2nd edition.

[16] An Introduction to Computing with Neural Nets (Richard P. Lipmann, IEEE ASSP Magazine, April 1987)

[17] Klimasauskas, CC. (1989). The 1989 Neuro Computing Bibliography. Hammerstrom, D. (1986). A Connectionist/Neural Network Bibliography.

[18] DARPA Neural Network Study (October, 1987-February, 1989). MIT Lincoln Lab.

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

[20] Ferry, G. (1987) "Networks on the Brain". *New Scientist.* 16 July 1987, pp54-58.

[21] Fahlman, S., 1988, Faster Learning Variations on Back Propagation: An Empirical Study: in Proceedings of the 1988 Connectionset Models Summer School. Hertz J.A., A.S. Krogh, and A. Palmer 1991, Introduction to the Theory of Neural Network.

[22] Lapedes A., and R. Farber. 1987. Nonlinear Signal Processing Neural Networks: Prediction and System Modeling. Technical Report: LA-UR-87-2662. Los Alamos National Laboratory, Los Alamos, NM 87545.

[23] White H. "Learning in Artificial Neural Networks: A statistical Perspective." 1989. Neural Computation. \FB1\fP, 425-464. MIT Press, Cambridge, MA.

[24] Data & Analysis Center for Software, "Artificial Neural Networks Technology", 1992 (http://www.dacs.dtic.mil/techs/neural/neural.title.html, printed November 1998)

[25] Fatemeh Zahedi, "Intelligent Systems for Business: Expert Systems with Neural networks, 1993 Wadsworth Inc. ISBN 0-534-18888-5.

[26] Moody, J. E. and Utans, J. (1994) 'Architecture Selection Strategies for Neural Networks: Application to Bond Rating Prediction', in Refenes, A. N. (ed.) *Neural Networks in Capital Markets* (New York, NY: John Wiley and Sons).

[27] Refenes, A. N. and Zaidi, A. (1992) 'Managing Exchange Rate Prediction Strategies with Neural Networks', in Lisboa, P. G., and Taylor, M. (ed.) *Neural Networks: Techniques and Applications* (Chichester: Ellis Horwood).

[28] White, Halbert (1988) 'Economic Prediction Using Neural Networks: The Case Of IBM Daily Stock Returns', Technical Report 88-20, University Of California (San Diego).

[29] Margarita, Sergio (1992a) 'Interacting Neural Networks: An Artificial Life Approach to Stock Markets', in Aleksander, Igor and Taylor, J. (eds.) *Artificial Neural Networks II: The Proceedings of the Second International Conference on Artificial Neural Networks (ICANN-92), 4-7 September, Brighton*, two volumes (Amsterdam: North-Holland).

[30] Odom, M. and Sharda, R. (1990) `A Neural Network Model for Bankruptcy Prediction', in Proceedings of the International Joint Conference on Neural Networks, San Diego, 17-21 June 1990, Volume II (Ann Arbor, MI: IEEE Neural Network Council), pp. 163-8.

[31] Krîse, Ben J. A. and van der Smagt, P. Patrick (1991) `An Introduction to Neural Nets', Draft Book, 4th Edition, Faculty of Mathematics and Computer Science, University of Amsterdam, September. [FTP] [EC] [DISCARDED].

[32] Data & Analysis Center for Software,"Artificial Neural Network Technology", 1992 (http://www.dacs.dtic.mil/techs/neural/neural.title.html, printed November 1998)

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

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