NEAR EAST UNIVERSITY.

Faculty of Engineering

Department of Computer Engineering

IMPLEMENTATIONS OF NEURAL NETWORKS

Graduation Project
COM-400

Student: Manaf Eloqlah

Supervisor: Assoc. Prof. Dr Adnan Khashman

Nicosia - 2003

ACKNOWLEDGMENT

"At this juncture, I express my deep sense of gratitude towards my project supervisor Assoc. Prof Dr Adnan Khashman for his constant inspiration, guidance and support throughout my project tenure. I consider myself very fortunate to be gifted with the golden opportunity of working under his guidance.

My parents and all family members have been a strong pillar of support for me and they have always encouraged me. My project wouldn't have seen the day's light without their constant motivation. I owe them a lot.

I wouldn't be able to express my feeling in words for my friends Yehia, Adham, Anas, Mohammad, Tarek, Bela/ and Ramadan, I will never forget the sweet memories I spent with them.

Project is such a function, which can be evaluated successfully, only if it is integrated between failure and success."

ABSTRACT

Work on artificial neural networks, commonly referred to as "neural networks", has been motivated right :from its inception by the recognition that the brain computes in an entirely different way from the conventional digital computer.

When we are talking about a neural network, we should more properly say "artificial neural network" (ANN), Biological neural networks are much more complicated than the mathematical models we use for artificial neural networks. But it is customary to be lazy and drop the "A" Of the "artificial"

Neural networks are computing devices that are loosely based on the operation of the brain. A neural network consists of a large number of simple processing units (Of "neurons"), massively interconnected and operating in parallel.

This project describes what artificial neural networks are, how to use them, and where they are currently being applied. A brief overview of neural networks and their history is provided. The project describes the individual neurons that comprise an artificial neural network in detail, along with the most common training procedures in use. Neural networks architectures and algorithms are presented. The project briefly summarizes application areas where neural networks are commonly applied. Finally, neural networks application in fraud detection is noted.

TABLE OF CONTENTS

INTRODUCTION

The human brain is the most elaborate information processing system known. In current thinking, it derives most of its processing power from the huge numbers of neurons and connections. There is no universally accepted definition of a neural network. But perhaps rnost people in the field would agrce that a neural network is a network of rnany sirnple processors ("units"), each possibly having a srnall arnount of local rnemory. The units are connected by communication channels ("connections") which usually carry numeric (as opposed to syrnbolic) <lata, encoded by any of various rneans. The units operate only on their local <lata and on the inputs they receive via the connections. The restriction to local operations is often relaxed during training.

Although Artificial Neural Networks (ANNs) have been around since the late 1950\s, it wasn't until the rnid-1980's that algorithms became sophisticated enough for general applications. Today ANNs are being applied to an increasing number of real-world problerns of considerable cornplexity.

There are rnultitudes of different types of ANNs. Some of the more popular include the multilayer perceptron which is generally trained with the back propagation of error algorithm, leaming vector quantization, radial basis function, Hopfield, and Kohonen, to name a few. Some ANNs are classified as feedforward while others are recurrent (i.e., irnplernentfeedback) depending on how <lata is processed through the network. Another way of classifying ANN types is by their rnethod of learning (or training), as sorne ANNs ernploy supervised training while others are referred to as unsupervised Of self-organizing. Supervised training is analogous to a student guided by an instructor. Unsupervised algorithms essentially perform clustering of the <lata into sirnilar groups based on the measured attributes or features serving as inputs to the algorithms. This is analogous to a student who derives the lesson totally on his Of her own. ANNs can be irnplernentedin software of in specializedhardware.

The airn of this project is to describe that process by explaining these structures, and to discuss what types of applications are currently utilizing the differenı structures and how sorne structures lend thernselvesto specific solutions.

Chapter one is presented as in introduction to Neural Networks and describes a general understanding of neural networks, the history of the work done in their field, how a real brain operates and how neural networks are currently being applied.

Chapter two describes various types of neural network structures. Single layer feedforward network, multilayer feedforward network, recurrent network and radial basic function are reviewed. Learning processes of neural networks including the supervised and unsupervised learning and algorithms used to train neural networks are described.

Chapter three surveys how artificial neural networks are being applied given applications of neural networks in some fields like medicine, business, image compression, language processing, character and pattern recognition, servo control, arts and telecommunication.

Chapter four shows how artificial neural networks used to develop a model to detect credit card fraud and how fraud detection is an important application of neural networks. Unsupervised learning methods and their applications to the fraud detection are described.

CHAPTERONE

INTRODUCTION TO NEURAL NETWORKS

1.1 Overview

This chapter intended to act a brief introduction to Artificial Neural Network technology and what Artificial Neural Networks are, how to use them, why they are important and who should know about Neural Networks. And will explain where Artificial Neural Networks have come from and presents a brief history of Neural Networks. Also this chapter discusses how they are currently being applied, and what types of application are currently utilizing the different structures. It will also detail why ~here has been such a large amount of interest generated in this are, and where the future ofthis technology may lie.

1.2 Artificial Neural Networks

Artificial Neural Networks are being touted as the wave of the future in computing. They are indeed self learning mechanisms which don't require the traditional skills of a programmer. But unfortunately, misconceptions have arisen. Writers have hyped that these neuron-inspired processors can do almost anything. These exaggerations have created disappointments for some potential users who have tried, and failed, to solve their problems with neural networks. These application builders have often come to the conclusion that neural networks are complicated and confusing.

Unfortunately, that confusion has come from the industry itself An avalanche of articles has appeared touting a large assortment of different neural networks, all with unique claims and specific examples. Currently, only a few of these neuron-based structures, paradigms actually, are being used commercially. One particular structure, the feedforward, backpropagation network, is by far and away the most popular. Most ofthe other neural network structures represent models for "thinking" that are still being evolved in the ll;\p9ratories.Yet, all of these networks are simply tools and as such the only real demand they make is that they require the network architect to learn how to use theri:ı.

1.3 Definition ofa Neural Network

Neural networks have a large appeal to many researchers due to their great closeness to the structure of the brain, a characteristic not shared by more traditional systems.

In an analogy to the brain, an entity made up of interconnected neurons, neural networks are made up of interconnected processing elements called units, which respond in parallel to a set of input signals given to each. The unit is the equivalent of its brain counterpart, the neuron.

A neural network consists offour main parts:

1. Processing units, where each unit has a certain activation level at any point in time.
2. Weighted interconnections between the various processing units which determine how the activation of one unit leads to input for another unit.
3. An activation rule which acts on the set of input signals at a unit to produce a new output signal, or activation.
4. Optionally, a learning rule that specifies how to adjust the weights for a given input/output pair.

One of the most important features ofa neural network is its ability to adapt to new environments. Therefore, learning algorithms are critical to the study of neural networks.

1.4 History of Neural Networks

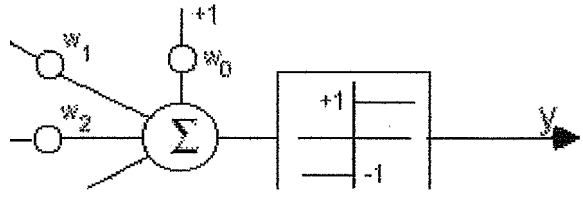
The study of the humarı brain is thousands of years old. With the advent of modern electronics, it was only natural to try to harness this thinking process.

The history of neural networks can be traced back to the work of trying to model the neuron. The first model of a neuron was by physiologists, McCulloch and Pitts (1943) [1]. The model they created had two inputs anda single output. McCulloch and Pitts noted that a neuron would not activate if only one of the inputs was active. The weights for each input were equal, and the output was binary, Until the inputs summed up to a certain threshold level, the output would remain zero. The McCulloch and Pitts' neuron has become known today as a logic circuit.

The *perceptron* was developed as the next model of the neuron by Rosenblatt (1958) [2], as seen in Figure 1.2. Rosenblatt, who was a physiologist, randomly

interconnected the perceptrons and used trial and error to randomly change the weights in order to achieve "Iearning." Ironically, McCulloch and Pitts' neuron is a much better model for the electrochemical process that goes on inside the neuron than the perceptron, which is the hasis for the modern day field of neural networks (Anderson and Rosenfeld, 1987) [3].

The electrochernical process of a neuron works tike a voltage-to-frequency translator (Anderson and Rosenfeld, 1987) [3]. The inputs to the neuron cause a chernical reaction such that, when the chemicals build to a certain threshold, the neuron discharges. As higher inputs come into the neuron, the neuron then fires at a higher frequency, but the magnitude of the output from the neuron is the sarne. Figure 1.2 is a model ofa neuron. A visual comparison of Figures 1.1 and 1.2 shows the origins of the idea of the perceptron can be traced back to the neuron. Externally, a perceptron seems to resemble the neuron with multiple inputs and a single output. However, this similarity does not really begin to model the complex electrochemical processes that actually go on inside a neuron. The perceptron is a very simple mathematical representation ofthe neuron.



X.S.~-1

Figure 1.1. The Perceptron

Selfödge (1958) [4] brought the idea of the weight space to the perceptron. Rosenblatt adjusted the weights in a trial-and-error method. Selfridge adjusted the weights by randomly choosing a direction vector. If the performance did not improve, the weights were returned to their previous values, and a new random direction vector was chosen. Selfridge referred to this process a,s climbing .tlıe mountain, as seen in Figur·e 1.3. Today, it is referred to as descending on the graqi~:nt because, generally, error squared, or the energy, is being rninirnized.

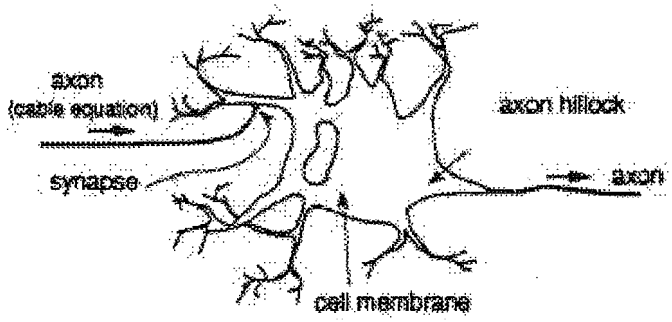


Figure 1.2. The Neuron

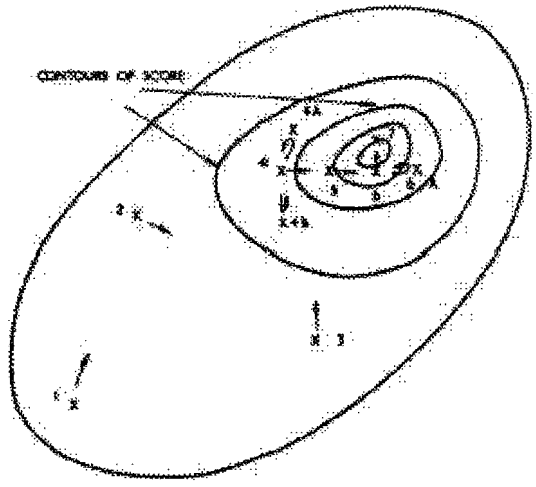


Figure 1.3. Climbing the Mountain

Widrow and Hoff (1960) [5] developed a mathematical method for adapting the weights. Assuming that a desired response existed, a gradient search method was implemented, which was based on minimizing the error squared. This algorithm would later become known as LMS, or Least Mean Squares. LMS, and its variations, has been used extensively in a variety of applications, especially in the last few years. This gradient search method provided a mathematical method for finding an answer that minimized the error. The learning process was not a trial-and-error process. Although the computational time decreased with Selfridge's work, the LMS method decreased the amount of computational time even more, which made use of perceptrons feasible.

At the height of neural network or perceptron research in the 1960's, the newspapers were full of articles promising robots that could think. It seemed that perceptrons could solve any problem. üne bqô~, Perceptrons (Minsky and Papert, 1969) [6], brought the research to an abrupt halt. The book points out that perceptrons could only solve linearly separable problems. A perceptron is a single node. Perceptrons

shows that in order to solve an n-separable problem, n-1 nodes are needed. A perceptron could then only solve a 2-separable problem, or a linearly separable problem.

After Perceptrons was published, research into neural networks went unfunded, and would remain so, until a method was developed to solve n-separable problems. Werbos (1974) [7] was first to develop the back propagation algorithm.

It was then independently rediscovered by Parker (1985) [8] and by Rumelhart and McClelland (1986) [9], simultaneously. Back propagation is a generalization of the Widrow-Hoff LMS algorithm and allowed perceptrons to be trained in a multilayer configuration, thus a n-1 node neural network could be constructed and trained. The weights are adjusted based on the error between the output and some known desired output. As the name suggests, the weights are adjusted backwards through the neural network, starting with the output layer and working through each hidden layer until the input layer is reached. The back propagation algorithm changes the schematic of the perceptron by using a sigmoidal function as the squashing function. Earlier versions of the perceptron used a signum function. The advantage of the sigmoidal function over the signum function is that the sigmoidal function is differentiable, This permits the back propagation algorithm to transfer the gradient information through the nonlinear squashing function, allowing the neural network to converge to a loca! minimum. Neurocomputing: Foundations of Research (Anderson and Rosenfeld, 1987) [3] is an excellent source of the work that was done before 1986. It is a coUection of papers and gives an interesting overview ofthe events in the fıeld ofneural networks before 1986.

Although the golden age of neural network research ended 25 years ago, the discovery of back propagation has reenergized the research being done in this area, The feed-forward neural network is the interconnection of perceptrons and is used by the vast majority ofthe papers reviewed.

1.5 What are Artificial Neural Networks?

Artificial Neural Networks are relatively crude electronic models based on the neural structure of the brain. The brain basicaliy leams from experience. It is natural proof that some problems that are beyond the scope of current computers are indeed solvableby small energy efficientpackages.

This brain modeling also promises a less technical way to develop machine solutions. This new approach to computing also provides a more graceful degradation during system overload than its more traditional counterparts.

These biologically inspired methods of computing are thought to be the next major advancement in the computing industry. Even simple animal brains are capable of functions that are currently impossible for computers.

Computers do rote things well, like keeping ledgers or performing complex matlı. But computers have trouble recognizing even simple patterns much less generalizing those patterns ofthe past into actions ofthe future.

Now, advances in biological research promise an initial understanding of the natural thinking mechanism. This research shows that brains store information as patterns. Some of these patterns are very complicated and allow us the ability to recognize individual faces :from many different angles.

This process of storing information as patterns, utilizing those patterns, and then solving problems encompasses a new field in computing. This field, as mentioned before, does not utilize traditional programming but involves the creation of massively parallel networks and the training of those networks to solve specific problems. This fıeld also utilizes words very different :from traditional computing, words like behave, react, self-organize, learn, generalize, and forget.

1.5.1 Analogy to the Brain

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

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

The individual neurons are complicated. They have a myriad of parts, sub-systems, and control mechanisms. They convey information via a host of electrochemical

pathways. There are over one hundred different classes of neurons, depending on the classification method used. Together these neurons and their connections form a process which is not binary, not stable, and not synchronous. In short, it is nothing like the currently available electronic computers, or even artificial neural networks.

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

1.5.2 Artificial Neurons and How They Work

The fundamental processing element of a neural network is a neuron. This building block of human awareness encompasses a few general capabilities. Basically, a biological neuron receives inputs :from other sources, combines them in some way, performs a generally nonlinear operation on the result, and then outputs the final result. Figure 1.4 shows the relationship of these four parts.

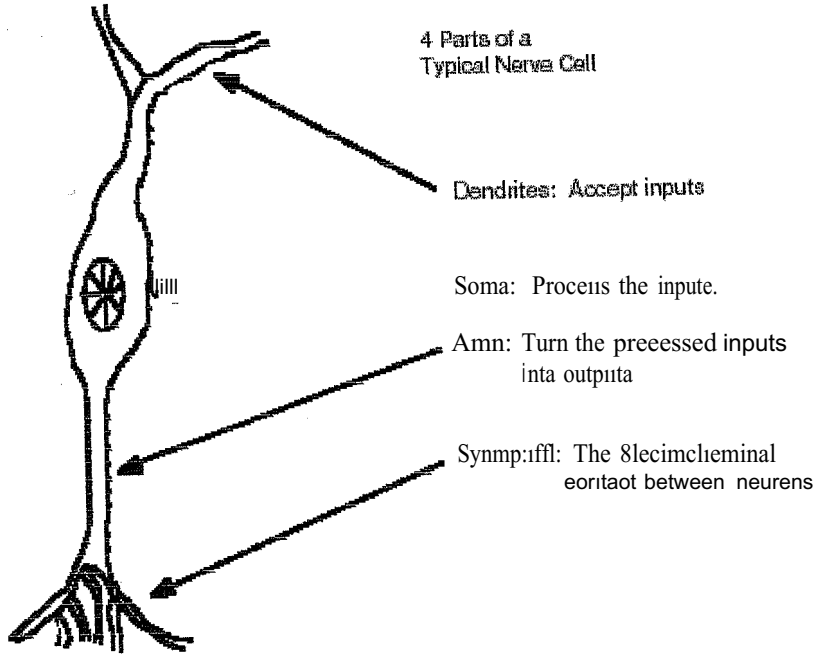


Figure 1.4. A Simple Neuron.

Within humans there are many variations on this basic type of neuron, further complicating man's attempts at electrically replicating the process of thinking. Yet, all natural neurons have the same four basic components.

These components are known by their biological names - dendrites, soma, axon, and synapses. Dendrites are hair-like extensions of the soma which act like input channels. These input ehannels receive their input through the synapses of other neurons. The soma then processes these incoming signals over time. The soma then turns that processed value into an output which is sent out to other neurons through the axon and the synapses.

Recent experimental <lata has provided further evidence that biological neurons are structurally more complex than the simplistic explanation above.

They are significantly more complex than the existing artificial neurons that are built into today's artificial neural networks. As biology provides a berter understanding of neurons, and as technology advances, network designers can continue to improve their systems by building upon man's understanding of the biological brain.

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

To do this, the basic units of neural networks, the artificial neurons, simulate the four basic functions of natural neurons. Figure 1.5 shows a fundamental representation of an artificial neuron.

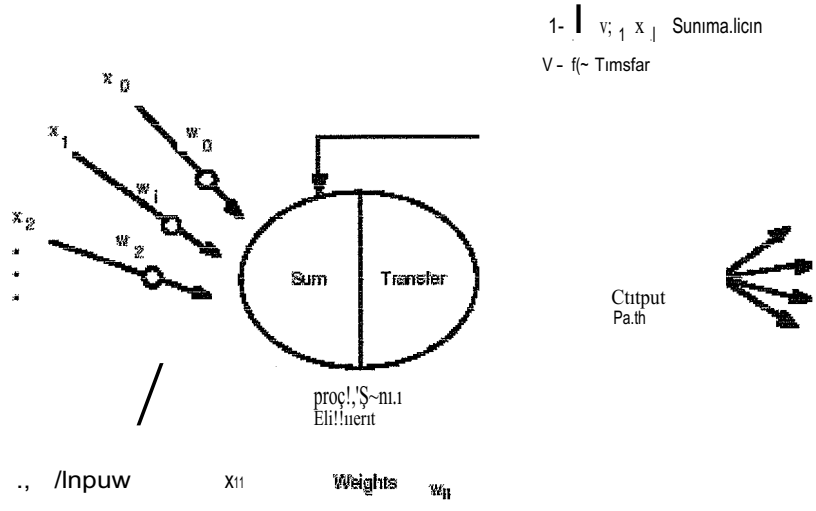


Figure 1.5. A Basic Artificial N\';)uron.

In Figure 1.5, various inputs to the network are t$presented by the :rpç'f,t;lıematical symbol, x(n). Each of these inputs is multiplied by a connection weight. Thl';lşe weights are represented by w(n). In the simplest case, these products are simply summed, fed through a transfer function to generate a result, and then output. This process lends itself

to physical implementation on a large scale in a small package. This electronic implementation is still possible with otlıer network structures which utilize different summing functions as well as different transfer functions.

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

Other networks work on problems where the resolutions are not just one of several known values. These networks need to be capable of an infinite number of responses. Applications of tlıis type include the "intelligence" belıind robotic movements. This "intelligence" processes inputs and then creates outputs wlıich actually cause some device to move.

That movement can span an infinite number of very precise motions. These jıetworks do indeed want to smooth tlıeir input wlıich, due to limitations of sensors, eomes in non-continuous bursts, say tlıirty times a second. To do tlıat, tlıey might accept ihese inputs, sum that data, and then produce an output by, for example, applying a hyperbolic tangent as a transfer functions. In tlıis manner, output values from the network are continuous and satisfy more real world interfaces,

Other applications might simply sum and compare to a threshold, .thereby producing one oftwo possible outputs, a zero ora one. Other functions scale tlıe outputs to match the application, such as the values minus one and one. Some functions even integrate tlıe input <lata over time, creating time-dependent networks.

1:6 Why Are Neural Networks Important?

Neural networks are responsible fq.ı;~the basic functions of our nervous system. The:y determine how we behave as an individual. Our emotions experienced as fear, anger, and what we enjoy in life come from ıı~ural networks in the brain. Even our abiıity to tlıink and store memories depends on neural networks. Neural networks in the

brain and spinal cord program all our movements including how fast we can type on a computer keyboard to how well we play sports. Our ability to see or hear is disturbed if something happens to the neural networks for vision or hearing in the brain,

Neural networks also control important :functionsof our bodies. Keeping a constant body temperature and blood pressure are examples where neural networks operate automatically to make our bodies work without us knowing what the networks are doing. These are called autonomic :functions of neural networks because they are automatic and occur continuously without us being aware ofthem.

I. 7 How Neural Networks Differ from Traditional Computing and Expert Systems

Neural networks offer a different way to analyze data, and to recognize patterns within that data, than traditional computing methods. However, they are not a solution for. all computing problems. Traditional computing methods work well for problems that can be well characterized. Balancing checkbooks, keeping ledgers, and keeping tabs of .inventory are well defined and do not require the special characteristics of neural networks. Table 1.1 identifies the basic differences between the two computing approaches.

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

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

Table 1.1. Comparison ofComputing Approaches.

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

Typically, an expert system consists of two parts, an inference engine and a knowledge base. The inference engine is generic. It handles the user interface, external files, program access, and scheduling. The knowledge base contains the information that is specific to a particular problem. This knowledge base allows an expert to define the rules which govern a process.

This expert does not have to understand traditional programming. That person simply has to understand both what he wants a computer to do and how the mechanism of the expert system shell works. It is this shell, part of the inference engine that actually tells the computer how to implement the expert's desires. This implementation oecurs by the expert system generating the computer's programming itself it does that through "programming" of its own. This programming is needed to establish the rules for a parıicularapplication. Tıtj,s method of establishing rules is also complex and does require ~>Jletail oriented person.

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

Artificial neural networks o:ffer a coı:upletelydit,ferentapproach to problem solving and they are sometimes called the sixth generation of computing. They try to provide a tool that both programs itself and learns on its own. Neur~l networks are structured to

provide the capability to solve problems without the benefits of an expert and without the need of programming. They can seek pattems in data that no one knows are there.

A comparison of arti:ficial intelligence's expert systems and neural networks is contained in Table 1.2.

Table 1.2 Comparisons ofExpert Systems and Neural Networks.

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

Expert systems have enjoyed significant succeşşys. However, artificial intelligence has encountered problems in areas such as vision, contimıous speech recognition and synthesis, and machine learning. Arti:ficialintelligence also is hostage to the speed of the processor that it runs on. Ultimately, it is restricted to the theoretical limit ofa single processor. Artiflcial intelligence is also burdened by the fact that experts don't always speakin rules.

Yet, despite the advantages of neural networks over both expert systems and more traditional computing in these specific areas, neural nets are not complete solutions. They offer a capability that is not ironclad, such as a debugged accounting system. They learn, and as such, they do continue to make "mistakes." Furthermore, even when a network has been developed, there is no way to ensure that the network is the optimal network.

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

A data set which includes the information which can characterize the problem.

An adequately sized <lata set to both train and test the network.

An understanding of the basic nature of the problem to be solved so that basic first-cut decision on creating the network can be made. These decisions include the activization and transfer functions, and the learning methods.

An understanding ofthe development tools.

Adequate processing power (some applications demand real-time processing that exceeds what is available in the standard, sequential processing hardware. The development ofhardware is the key to the future ofneural networks).

ünce these conditions are met, neural networks offer the opportunity of solving problems in an arena where traditional processors lack both the processing power and a step-by-step methodology. A number of very complicated problems cannot be solved in the traditional computing environments. For example, speech is something that ali people can easily parse and understand. A person can understand a southern drawl, a Bronx accent, and the slurred words of a baby. Without the massively paralleled processing power of a neural network, this process is virtually impossible for a computer. Image recognition is another task that a human can easily do but which stymies even the biggest of computers. A person can recognize a plane as it turns, flies overhead, and disappears into a dot. A traditional computer might try to compare the changing images to a number of very differeı;ıt.st,ored patterns.

This new way of computing requires l̈j̇~j̇lş beyond traditional computing. It is a natura! evolution. Initially, computing was orily hardware and engineers made it work. Then, there were software specialists - programmeg; systems engineers, <lata base specialists, and designers. Now, there are also neural aı-chit,ects. This new professional needs to be skilled different than this predecessors of the past. For instance, he will need

to know statistics in order to choose and evaluate training and testing situations. This skill of making neural networks work is one that will stress the logical thinking of current software engineers.

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

1.8 Who Should Know About Neural Networks?

Workers in areas dealing with people's health must understand neural networks. Doctors and nurses must understand them in order to take care of their patients. Paramedics (firefighters and ambulance teams) need the knowledge to make quick decisions for saving the lives of accident victims. Doctor's assistants in many different areas of special medical treatment use their understanding of the nervous system to do their jobs. Scientists must know what is already known in order to design studies that will produce new knowledge of how the nervous system works and new ways to treat diseases of the nervous system. *Finally, it is important for each of us to understand how our own bodies work.*

1.9 Neural Networks and Their Use

Neural networks are computing devices that are loosely based on the operation of the brain. A neural network consists of a large number of simple processing units (or "neurons"), massively interconnected and operating in parallel. In the brain's neocortex, there are about 10 billion neurons, each of which communicates with roughly 10 thousand others. Far more modest, a typical artificial neural network might have several hundred processing units and several thousands of interconnections.

The field of neural networks has experienced rapid growth in recent years and is now enioying an explosion of notoriety. The interest in neural networks stems from the çlaim that they can learn to perform a task based on examples of appropriate behavior. That is, rather than being programmed to perform a task, tike an ordinary computer, the neural network can program itself based on examples provided by a teacher. Neural networks have been widely applied to pattern recognition problems, including speech and printed character recognition, medical diagnosis, robotic control, and economic forecasting.

The idea of a neural network has broad intuitive appeal -- a computer built like the brain. In reality, there is nothing magical about neural networks. At a formal level, neural networks are primarily a set of tools and statistical techniques for nonlinear regression.

Many of these techniques have been around for a long time under different names and in different fields, but the :field of neural networks has helped to unify them. Most importantly, these techniques had not previously been applied to problems in artificial intelligence, machine learning, and adaptive control.

First generation products developed by Sensory used neural networks för sound classification. A spoken word is given as input, and the network's task is to classify the sound as one of the possible words within a set of words. The network is trained by a supervised learning paradigm: it is provided with a set of examples of categorized sounds, and the connection strengths between units are adjusted to produce the appropriate category response to each of the training examples. If the network has learned well, it will then be able to generalize -- i.e., correctly classify new examples of the words. (Sensory's current generation of products use two other training paradigms: unsupervised and reinforcement-based methods. Unsupervised methods discover regularities in the environment; reinforcement-based methods involve learning from rewards and punishment.)

The art of neural network design involves specifying three key elements: the neural network architecture, the input and output representations, and the training method. The architecture defines the connectivity of the processing units and their dynamics - how one unit affects the activity of another. The input and output representations encode the information (e.g., words, patterns) fed into and read from the net in terms of a pattern of neural activity (numerical vectors). The training method specifies how the connection strengths in the network are determined from the <lata; the method includes techniques for comparing alternative models and for verifying the quality of the resulting network.

The success or failure of the neural network is deeply rooted in the appropriate selection of the above elements. Commercial software that simulates a neural network is unlikely to provide the optjm,µnı sohıtion; an experienced practitioner is required to tailor the three key elements to the application domain, For example, in our research, an off-the-shelf neural network tested on a spoken digit recognition problem correctly recognizes only 80% of words across speakers. Using Sensory's neural network architecture, I/0 representations, and training methods, performance jumps to over 95%

correct recognition. Performance further increases to over 99% correct with the application of additional proprietary techniques developed by Sensory.

The details of the Sensory neural network speech recognizer are company trade secrets and are the subject of patent applications, They involve preprocessing of the raw acoustic signal into a rate and distortion-independent representation that is fed into the neural network. The neural network is structured to perform nonlinear Bayesian classification. Because of an explicit probabilistic model in the network, prior class probabilities can be incorporated and the network outputs can be interpreted as a probability distribution over classes. The training procedure explores the space of neural network models as well as weighting coefficients; cross-validation techniques are used for model selection. Training data consists ofa large corpus of 300-600 voice samples representative of potential application users.

1.10 Where are Neural Networks being used?

Neural networks are being used in numerous applications in business and industry. Because neural networks can identify complex patterns and trends, known as pattern recognition, in data, they are useful in applications for data mining, classification, forecasting and prediction. The neural network can sort through an extensive amount of data performing the function much as a humarı would if the human were able to analyze the same amount of information in a short time.

Data mining involves processing massive amounts of information to identify key factors and trends. This information can then be used to improve managerial decision-making. As is described in our textbook, Bank of America uses data mining software to develop strategies for cultivating the most profitable customers. Customers who were likely to purchase a high margin lending product were identified by examining Bank of America's database. On the order ofthree hundred <lata points for each customer in the database were examined and neural networks identified those customers who would be interested in this type of loan. Another example of data mining is in the field of marketing. Neural networks are used to identify consumer profiles based on web-surfing histories, to enhance targeted marketing.

An example of the use of neural networks for classification is in computer aided diagnosis (CAD) in mammography for detection of breast cancer. The goal of CAD is to assist the radiologist in the screening process to provide the most accurate diagnosis

for the least investment of the professional 's time. Microcalcifications are seen as small bright spots in mammograms. They are clinically relevant and differ from other normal structures when they appear in specific types of clusters or pattems. The digitized image is processed to reveal microcalcifications. The neural network is then employed to review the patterns of microcalcifications to identify potential breast cancer. Other areas in which neural networks can be used for classification are signature verification, loan risk evahıation, voice recognition, and property valuation.

Neural networks are also being used in forecasting and prediction. üne example of the use for prediction is in emergency room triage. Based on pattern recognition the neural network can prioritize the patients for the most efficient utilization of resources. Another area in which neural networks can be used for forecasting is in investment analysis. Based on analysis of historical patterns the neural network can predict the movement of securities in the market. Other areas in which neural networks are used for forecasting and prediction include economic indicators, crop forecasts, weather forecasts, future sales levels and even the outcome of sporting events such as horse racing and baseball.

Organizations today are faced with increasing amounts of data. Neural networks provide an expedient and powerful solution for analyzing the <lata to assist in decision making.

1.11 The Future of Artificial Neural Networks

There is no doubt that neural networks are here to stay. There has been an intense amount of interest in them during recent years and as technology advances they will only become more valuable tools. There are a number of potential avenues that have, as yet, remain untapped, that will help to bring this technology to the forefront.

The :first of these is the development of hardware acceleration for neural circuitry. It has been shown time and time again that when a technology begins to be supported by dedicated hardware that advances come in leaps and bounds. At present much of the work undertaken is done via so:ftware simulation, which obviously places severe restrictions upon performance.

Clearly the problem domain is going to dictate the speeds of execution needed, with only the most demanding (e.g. real-time) requiring dedicated hardware support. Stili there is going to be a great demand for this technology and as such many

semiconductor companies are now developing VLSI chips with neural applications in mind. Many of these chips are designed with end user programmirıg abilities (FPGA), which pemrits designs to be rapidly tested at a low cost.

There are a great many areas of application that will demand the highest levels of performance and therefore hardware acceleration. For example military applications require rugged and reliable systems that are capable of high performance in difficult conditions. The prospect of devices that can adapt to rapidly changing and adverse environments is a very attractive one. Medical, communications and control applications can all benefit from the increased performance afforded by hardware implementations Examples include the 'Electronic Nose', the diagnosis of heart defects from ECG traces and the filtering of EMG signals in order to operate actuators.

Some people argue that the gains in performance from hardware implemenrations are not so important because of the rapidly increasing power of standard processors. They argue that simulations will be just as fast within a year or two. This may well be true, but what must be remembered is that the technological advances that have led to the speed increases in conventional processors can also be applied to neural chips. Things in the computer world never stand still.

One of the next steps in the development of this technology is to produce machines capable of higher levels of cognition. At present neural systems have no real claim to any such abilities; the best that they can claim is to be on a par with our own pre-attentive processing.

There is a great desire for fully autonomous intelligent systems as there are many real applications just waiting for the right technology to come along. A good example of this planetary exploration which at present relies on remotely controlled devices. If you were able to land a probe on to a planets surface, with the task of surveying as much of the surrounding area as possible, without the need of explicit instruction or control, much more could be achieved than is currently possible.

There are many ways to try and attack this very complex problem, the first of which is to-build neural models ofpa:ı.-ticular brain.centres without to much regard to the underlying neural structure. The mqçlels can thenJ)~ tested against various behavioral paradigms like operant conditioning or classical conditioning, This technique is growing in its popularity with many neurophysiologists. An alternative approach is to attempt to replicate as much of the neural substructure's complexity as possible. The problem with this approach is that you very rapidly run out of availableprocessing power.

It is the modeling of particular brain centres that has been used in a European research project PSYCHE. Thls project attempts to relate results from non-invasive instruments (EEG and MEG) whlch measure the average neural activity over tens of thousands of neurons during information processing activities. The plan is to start from a simple model of average activity and then build up to a more complex one. The initial stages have simple neurons and no more than a hundred modules hard-wired together. Then more complexity and leaming will be applied to the model, whilst being constantly checked against experiment results to make sure that it is on the right track. As all of this is taking place the emergence of cognitive powers will be monitored using various psychological paradigms to provide more guidance to PSYCHE's learning.

Clearly artificial neural networks still have a long way to go before we start seeing the incredible creations of science fiction, but still they have achieved a lot in their relatively short history. They are now being used in many different areas with great success and continued research and development is going ensure that they become a more important part of our lives all the time.

1.12 Summary

In this chapter we have demonstrated a basic introduction to neural networks. Withln the chapter we have explained that neural networks are groups of select neurons that are connected with one another and they are functional circuits in the brain that process information and create useful activities by sending outputs to the body. As we have discussed in the sections, neural networks have had a unique history in the realm of technology and the earliest work in neural computing goes back to the 1940s when the first neural network computing model was developed. Also we have explained the definition of artificial neural networks as computing devices that are loosely based on the operation of the brain. Also we have considered the importance of neural networks, who should know about neural networks and their use. We have also explained where neural networks are being used giving some application of there use. Last but not least we have discussed the future of neural networks considering that there is a great deal of researches is going on in neural networks worldwide,

CHAPTERTWO

NEURAL NETWORKS ALGORITHMS

2.1 Overview

This Chapter presented a description of architectures and algorithms used to train neural networks. This chapter will explain the Model of a neuron and structures of neural networks inchıding the single layer feedforward networks, multilayer feedforward networks, recurrent networks, and radial hasis function networks. The sections below explains the artificial neural networks training and learning involved neural networks leaming; supervised leaming and unsupervised leaming. Also this chapter discusses some advanced neural networks learning and problems using neural networks.

2.2 Models ofa Neuron

A *neuron* is an information-processing unit that is fundamental to the operation ofa neural network. Figure 2.1 shows the model for a neuron. We may identify three basic elements of the neuron model, as described here:

1. A set of *synapses* or *connecting links,* each of which is characterized by a *weight* or *strength* of its own. Specially, a signal x_j at the input of synapse of j connected to neuron k is multiplied by the synaptic weight w_{iq}. It is important to make a note of the manner in which the subscripts of the synaptic weight w_{ki} are written. The first subscript refers to the neuron in question and the second subscript refers to the input end of the synapse to which the weight refers; the reverse of this notation is also used in the literature. The weight w_{ki} is positive if the associated synapse is excitatory; it is negative ifthe synapse is inhibitory.
2. An *adder* for summing the input signals, weighted by the respective synapses of the neuron; the operations described here constitutes *a liner combiner.*
3. An *activation function* for limiting the amplitude of the output of a neuron. The activation fiınction is also referred to in the literature as a *squashing function* in that it squashes (limits) the permissible amplitude range of the output signal to some finite value. Typically, the normalized amplitude range of the output of a neuron is written as the closed unit interval [0, 1] or alternatively [-1, 1].

The model of a neuron shown in Fig. 2.1 also includes an externally applied *threshold* θ_k that has the effect of lowering the net input of the activation function. On the other hand, the net input of the activation function may be increased by employing a *bias* term rather than a threshold; the bias is the negative of the threshold.

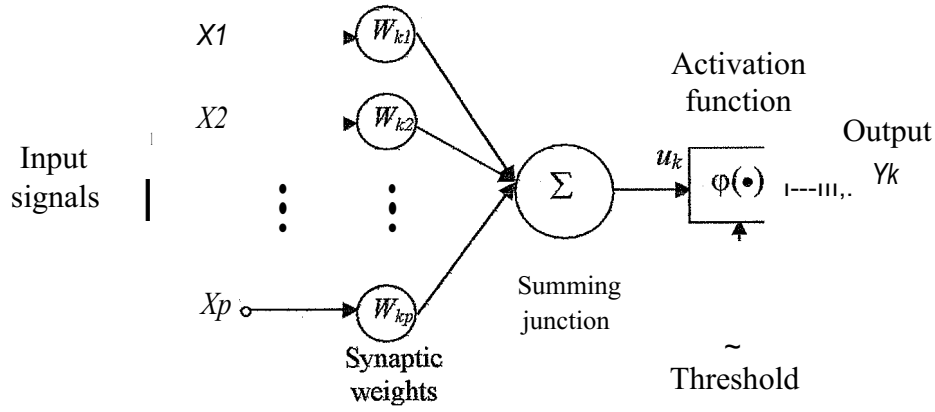


Figure 2.1 Nonlinear model ofa neuron.

In mathematical terms, we may describe a neuron k by writing the following pair of equations:

$$u_k = \sum_{J=1}^{p} fl'_{kJ} X_J \tag{2.1}$$

And

$$y_k = \varphi(u_k - \theta_k) \tag{2.2}$$

Where x1, x2, ..., xp are the input signals; Wk1, Wk2, ..., Wkp are the synaptic weights of neuron k; u_k is the *linear combiner output,* e_k is the *threshold;* $q(\cdot)$ is the *activation function;* and J_k is the output signal of the neuron. The use of thresholde1 has the effect of applying an *a.ffine transformation* to the output u_1 of the linear combiner in the model of Fig 2.2 as shown by

$$v_k = u_k - \theta_k \tag{2.3}$$

In particular, depending on whether the thfy~hold e_k is positive of negative, the relationship between the effective internal *actiiJit)!.fevel* or *activation potential* v_k of neuron k and the linear combiner output u_k is modified i1:1\the manner illustrated in Fig. 2.2. Note that asa result ofthis a:ffine transformation, the graph of vs versus u_{lc} no longer pass through the origin.

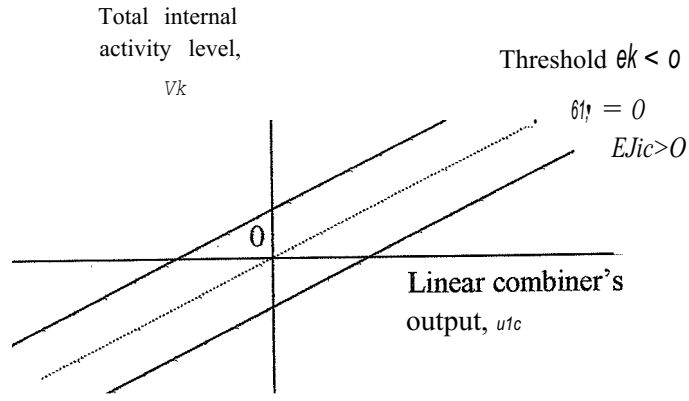


Figure 2.2. Affine transformation produced by the presence ofa threshold.

The B_{IC} is an external parameter of artificial neuron k. We may account for its presence as in Eq. (2.2). Equivalently, we may formulate the combination of Eqs. (2.1) and (2.2) as follows:

$$vk = \sum_{j=0}^{p} HiJXJ \tag{2.4}$$

and

$$Yk = tp(vk) \tag{2.5}$$

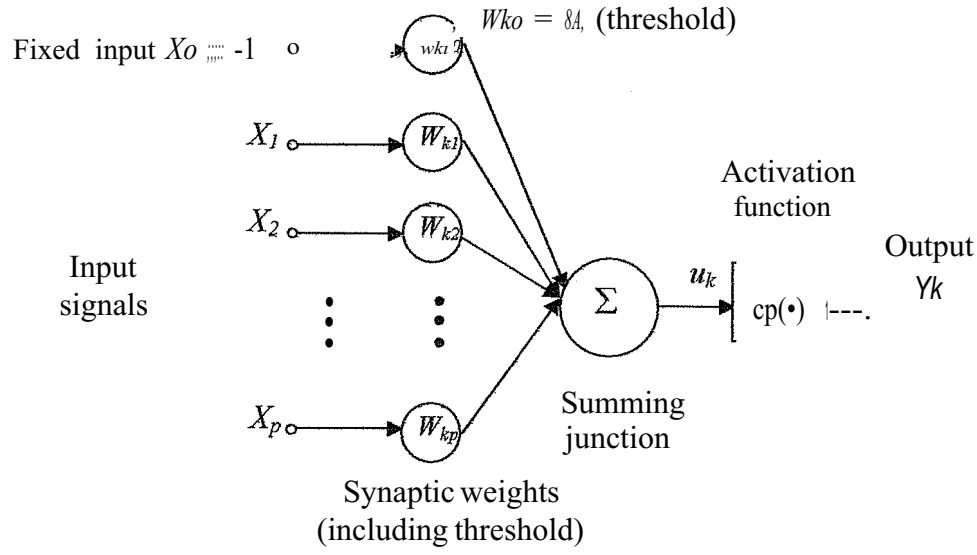
In Eq. (2.4) we have added a new synapse, whose input is

$$X_a = -1 \tag{2.6}$$

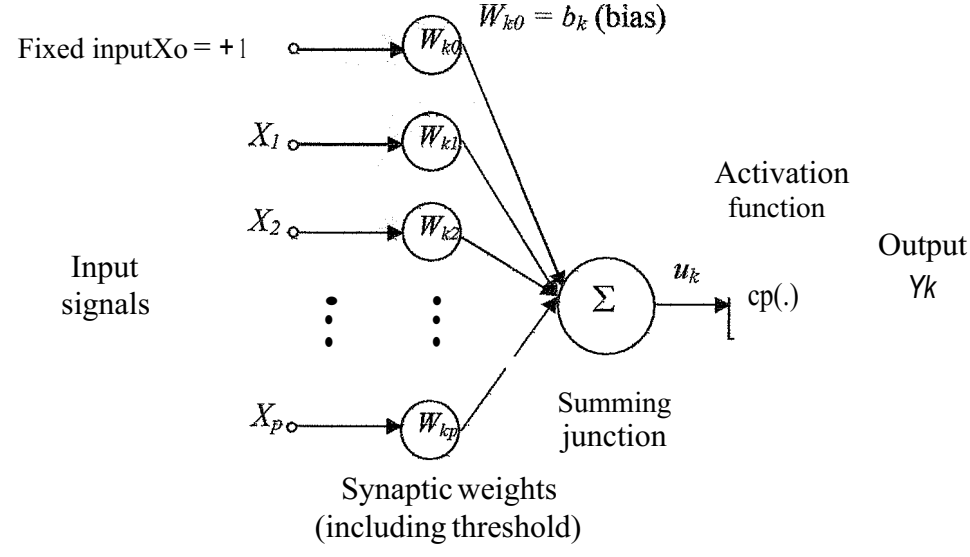
and whose weight is

$$W_{k0} = \theta_k \tag{2.7}$$

We may therefore reformulate the model of neuron kas in Fig. 2.3a. In this figure, the effect of the threshold is represented by doing two things: (1) adding a new input signal fixed at -1, and (2) adding a new synaptic weight equal to the threshold B_{IC}. Alternatively, we may model the neuron as in Fig. 2.3b,



(a)



(b)

Figure 2.3. Two other nonlinear models ofa neuron.

Where the combination of fixed input *xo* = +1 and weight *wlco* = *blc* accounts for the *bias bı.* Although the models in Fig. 2.1 and 2.3 are different in appearance, they are mathematically equivalent,

2.3 Neural Network Struetures

The manner in which the neurons of a neural network are structured is intimately Iinked with the learning algorithm used to train the network. We may therefore speak of learning algorithms (rules) used in the design of neural networks as being structured.

In general, we may identify four different classes ofnetwork architectures:

2.3.1 Single-Layer Feedforward Networks

A *layered* neural network is a network of neurons organized in the form of layers. In the simplest form of a layered network, we just have an *input layer* of source nodes that projects onto an *output layer* of neurons (computation nodes), but not vice versa. In other words, this network is strictly of *afeedforwardtype.* It is illustrated in Fig. 2.4 for the case of four nodes in both the input and output layers. Such a network is called a *single-layer network,* with the designation "single layer" referring to the output layer of computation nodes (neurons). In other words, we do not count the input layer of source nodes, because no computation is performed there.

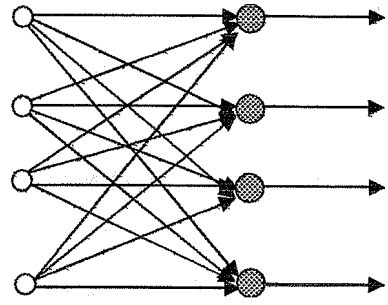


Figure 2.4. Feedforward network with a single layer ofneurons

Algorithm

The perceptron can be trained by adjusting the weights of the inputs with Supervised Learning. In this learning technique, the patterns to be recognised are known in advance, and a training set of input values are already classified with the desired output. Before commencing, the weights are initialised with random values. Each training set is then presented for the perceptron in turn. For every input set the output :from the perceptron is compared to the desired output. If the output is correct, no weights are altered. However, if the output is wrong, we have to distinguish which of the patterns we would like the result to be, and adjust the weights on the currently active inputs towards the desired result.

Perceptron Convergence Theorem:

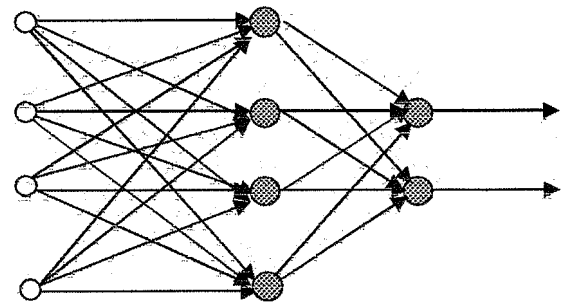
Theperceptron algorithm finds a linear discrtminant functtoninfinite iterations if the training set is linearly separable. [Rosenblatt 1962] [2].

The learning algorithm for the perceptron can be improved in several ways to improve e:fficiency, but the algorithm lacks usefulness as long as it is only possible to classify linear separable patterns.

2.3.2 Multilayer Feedforward Networks

The second class of a feedforward neural network distinguishes itself by the presence of one or more *hidden layers,* whose computation nodes are correspondingly called *hidden neurons* or *hidden units.* The function of the hidden neurons is to intervene between the external input and the network output. By adding one or more hidden layers, the network acquires a *global* perspective despite its local connectivity by virtue of the extra set of synaptic connections and the extra dimension of neural interactions (Churchland and Sejnowski, 1992) [10]. The ability of hidden neurons to extract higher-order statistics is particularly valuable when the size of the input layer is large.

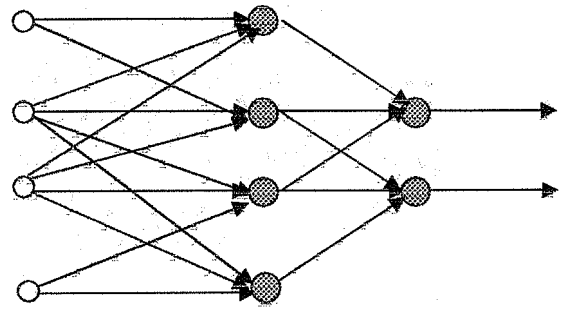
The source nodes in the input-layer ofthe network supply respectively elements ofthe activation pattern (input vector), which constitute the input signals applied to the neurons (computation nodes) in the second layer (i.e., the first hidden layer). The output signals ofthe second layer are used as inputs to the third layer, and so on for the rest of the network. Typically, the neurons in each layer of the network have as their inputs the output signals ofthe preceding layer only. The set of output signals ofthe neurons in the output (final) layer of the network constitutes the overall response of the network to the activation pattern supplied by the source nodes in the input (first) layer. The architectural graph of Fig. 2.5 illustrates the layout of a multilayer feedforward neural network for the case ofa single hidden layer. For brevity the network of Fig. 2.5 is referred to as a 4-4-2 network in that it has 4 source nodes, 4 hidden nodes, and 2 output nodes. As another example, a feedforward network with p source nodes, h_1 neurons in the first hidden layer, $h2$ neurons in the second layer, and q neurons in the output layer, say, is referred to asa p-$h1$-hrq network.



Input layer of source nodes | Hidden layer of neurons | Output layer of neurons

Fignre 2.5. Fully connected feedforward network with one hidden layer.

The neural network of Fig 2.5 is said to be *fully connected* in the sense that every nede in each layer of the network is connected to every other nede in the adjacent forward layer. If, however, some of the communication links (synaptic connections) are missing from the network, we say that the network is *partially cannected.* A form of partially connected mıltilayer feedforward network of particılar interest is a locally connected network. An example of such a network with a single hidden layer is presented in Fig. 2.6. Each neuron in the hidden layer is connected to a local (partial) set of source nodes that lies in its immediate neighborhood; sıch a set of Iocalized nodes feeding a neuron is said to constitute the *receptive field* of the neuron. Likewise, each neıron in the oıtpıt layer is connected to a local set of hidden neırons. The network of Fig. 2.6 has the same number of source nodes, hidden nodes, and outpıt nodes as that of Fig.2.1. However, comparing these two networks, we see that the locally connected network ofFig. 2.6 has a *specialized* structure.



Input !ayer of source nodes | Hidden layeı' of neurons | OıtputJayer of neurons

Figure 2.6. Partially connected feedforward network.

Algorithm

The threshold function of the units is modified to be a function that is continuous derivative, the Sigmoid Function. The use of the Sigmoid function gives the extra information necessary for the network to implement the back-propagation training algorithm. Back-propagation works by finding the squared error *(the Error function)* of the entire network, and then calculating the error term for each of the output and hidden units by using the output from the previous neuron layer. The weights of the entire network are then adjusted with dependence on the error term and the given learning rate. Training continues on the training set until the error function reaches a certain minimum. If the minimum is set too high, the network might not be able to correctly classify a pattern. But if the minimum is set too low, the network will have difficulties in classifying noisy patterns.

2.3.3 Recurrent Networks

A *recurrent neural network* distinguishes itself from a feedforward neural network in that it has at least one *feedforward* loop. For example, a recurrent network may consist of a single layer of neurons with each neuron feeding its output signal back to the inputs of all the other neurons, as illustrated in the architecture graph of Fig. 2.7. .In the structure depicted in this figure there are no self-feedback loops in the network; *self-feedback* refers to a situation where the output of a neuron is fed back to its own input. The presence of feedback loops has a profound impact on the learning capability of the network, and on its performance. Moreover, the feedback loops involve the use of particular branches composed of *unit-delay elements (*denoted by *:f1),* which result in a nonlinear dynamical behavior by virtue of the nonlinear nature of the neurons. Nonlinear dynamics plays a key role in the storage function of a recurrent network.

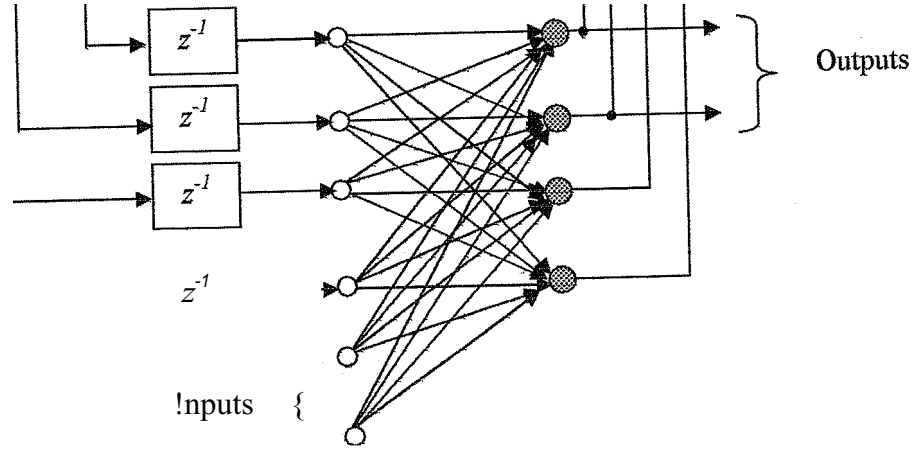


Figure 2.7. Recurrent network with hidden neurons.

2.3.4 Radial Basis Function Networks

The radial basis function (RBF) network constitutes another way of implementing arbitrary input/output mappings. The most significant difference between the MLP and RBF lies in the processing element n:onlinearity.While the processing element in the MLP responds to the full input space, the processing element in the RBF is local, normally a Gaussian kernel in the input space. Hence, it only responds to inputs that are close to its center; i.e., it has basically a *local response*.

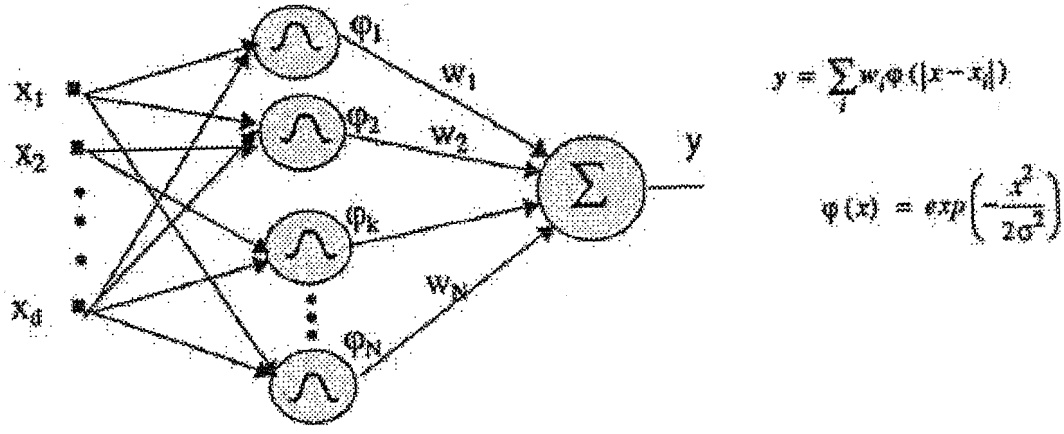


Figure 2.8. RadtçılBasis Funetioa (RBF) network.

The RBF network is also a layered net with the hi,~~en layer built from Gaussian kernels anda linear (or nonlinear) output layer (Fig. 2.8), Training of the RBF network is done normally in two stages [Haykin, 1994] [11]:

First, the centers *xi* are adaptively placed in the input space using competitive learning or *k* means clustering [Bishop, 1995] [12], which are unsupervised procedures. Competitive learning is explained later in the chapter. The variances of each Gaussian are chosen as a percentage (30 to 50%) to the distance to the nearest center. The goal is to cover adequately the input <lata distribution. ünce the RBF is located, the second layer weights *wi* are trained using the LMS procedure.

RBF networks are easy to work with, they train very fast, and they have shown good properties both for function approximation as classification. The problem is that they require lots of Gaıssian kernels in high-dimensional spaces.

2.4 Training an Artificial Neural Network

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

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

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

2.4.1 Supervised Training

In supervised training, both the inputs and the outputs are provided. The network then processes the inputs and compares its resulting outputs against the desired outputs. Errors are then propagated back through the system, causing the system to adjust the weights which control the network.

This process occurs over ang 9yer as the weights are continually tweaked. The set of data which enables the training fa called the "training set." During the training of a network the same set of data is processed ıma:ıy times as the connection weights are ever refined.

The current commercial network development packages provide tools to monitor how well an artificial neural network is converging on the ability to predict the right answer. These tools allow the training process to go on for days, stopping only when the system reaches some statistically desired point, or accuracy. However, some networks never learn. This could be because the input <lata does not contain the specific information from which the desired output is derived. Networks also don't converge if there is not enough <lata to enable complete learning. Ideally, there should be enough <lata so that part of the <lata can be held back as a test. Many layered networks with multiple nodes are capable of memorizing <lata. To monitor the network to determine if the system is simply memorizing its <lata in some nonsignificant way, supervised training needs to hold back a set of <lata to be used to test the system after it has undergone its training. (Note: memorization is avoided by not having too many processing elements.).

If a network simply can't solve the problem, the designer then has to review the input and outputs, the number of Iayers, the number of elements per layer, the connections between the layers, the summation, transfer, and training fiinctions, and even the initial weights themselves, Those changes required to create a successful network constitute a process wherein the "art" ofneural networking occurs.

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

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

When finally the system has been correctly trained, and no further leaming is needed, the weights can, if desit~d, be "fro1;~~,II In some systems this finalized network is then turned into hardware so th.at it can be fast. Other systems don't lock themselves in but continue to learn while in production use.

2.4.2 Unsupervised **Training**

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

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

One of the leading researchers into unsupervised learning is Tuevo Kohonen [13] an electrical engineer at the Helsinki University of Technology. He has developed a self-organizing network, sometimes called an autoassociator that learns without the bene:fit of knowing the right answer. It is an unusual looking network in that it contains one single layer with many connections. The weights for those connections have to be initialized and the inputs have to be normalized. The neurons are set up to compete in a winner-take-all fashion.

Kohonen continues his research into networks that are structured differently than standard, feedforward, back-propagation approaches. Kohonen's work deals with the grouping of neurons into fields. Neurons within a field are "topologically ordered." Topology is a branch of maıhematics that studies how to map from one space to another without changing the geometric configuration. The three-dimensional groupings often found in mammalian brains are an example of topological ordering. Kohonen has pointed out that the lack of topology in neural network models make today's neural networks just simple abstractions of the real neural networks within the brain. As this research continues, more powerful self learning networks may become possible. But eurrently, this field remains one that is still in the laboratory.

2.5 Teaching an Artifieial Neural Network

2.5.1 Supervised Learnlng

The vast majority of artificial neural network solutions have been trained with supervision. In this mode, the actual output of a neural network is compared to the desired output. Weights, which are usually randomly set to begin with, are then adjusted by the network so that the nex.t iteration, or eyde, will produce a closer match between the desired and the actual output. The learning method tries to minimize the current errors of all processing elements. This global error reduction is created over time by continuously modifying the input weights until an acceptable network accuracy is reached.

With supervised learning, the artificial neural network must be trained before it becomes useful. Training consists of presenting input and output <lata to the network. This <lata is often referred to as the training set. That is, for each input set provided to the system, the corresponding desired output set is provided as well. In most applications, actual <lata must be used. This training phase can consume a lot of time. In prototype systems, with inadequate processing power, leaming can take weeks. This training is considered complete when the neural network reaches a user defined performance level. This level signifies that the network has achieved the desired statistical accuracy as it produces the required outputs for a given sequence of inputs. When no further learning is necessary, the weights are typically frozen for the application. Some network types allow continual training, at a much slower rate, while in operation, This helps a network to adapt to gradually changing conditions.

Training sets need to be fairly large to contain all the needed information if the network is to leam the features and relationships that are important. Not only do the sets have to be large but the training sessions must include a wide variety of <lata. If the network is trained just one example at a time, all the weights set so meticulously for one fact could be drastically altered in learning the next fact. The previous facts could be forgotten in learning something new. As a;;ı.;esult, the system has to leam everything together, finding the best wefg~t settirlgs for the total set of facts. For example, in teaching a system to recognii~V,pixelpatt~,r)~9~{the ten digits, if there were twenty examples of each digit, ali the examples of the>digitseven should not be presented at the same time.

How the input and output <lata is represented, or encoded, is a major component to successfully instructing a network. Artificial networks only deal with numeric input <lata. Therefore, the raw <lata must often be converted from the external environment. Additionally, it is usually necessary to scale the <lata, or normalize it to the network's paradigm. This pre-processing of real-world stimuli, be they cameras or sensors, into machine readable format is already common for standard computers. Many conditioning techniques which directly apply to artificial neural network implementations are readily available. It is then up to the network designer to find the best data format and matching network architecture for a given application.

After a supervised network performs well on the training data, then it is important to see what it can do with <lata it has not seen before. If a system does not give reasonable outputs for this test set, the training period is not over. Indeed, this testing is critical to insure that the network has not simply memorized a given set of <lata but has learned the general patterns involved within an application,

2.5.2 Unsupervised **Learning.**

Unsupervised learning is the great promise of the future. It shouts that computers could someday learn on their own in a true robotic sense. Currently, this learning method is limited to networks known as self-organizing maps. These kinds of networks are not in widespread use. They are basically an academic novelty. Yet, they have shown they can provide a solution in a few instances, proving that their promise iş not groundless. They have been proven to be more effective than many algorithmic techniques for numerical aerodynamic flow calculations. They are also being used in the lab where they are split into a front-end network that recognizes short, phoneme-like fragments of speech which are then passed on to a backend network. The second artificial network recognizes these strings of fragments as words,

This promising field of unsupervised learning is sometimes called self-supervised learning. These networks use no external influences to adjust their weights. Instead, they internally monitor their performance. Tlıe,se networks look for regularities or trends in the input signals, and makes 3:9Jıptş,tion.5:.;)ş,ç9ording to the function of the network. Even without being told whether it's right or wtoıı g, the network still must have some information about how to organize itself This information is built into the network topology and learning rules.

An unsupervised Iearning algorithm might emphasize cooperation among clusters of processing elements. In such a scheme, the clusters would work together. If some external input activated any node in the cluster, the cluster's activity as a whole could be increased. Likewise, if external input to nodes in the cluster was decreased, that could have an inhibitory effect on the entire cluster.

Competition between processing elements could also form a basis for learning, Training of competitive clusters could arnplify the responses of specific groups to specific stimuli. As such, it would associate those groups with each other and with a specific appropriate response. Normally, when competition for learning is in effect, only the weights belonging to the winning processing element will be updated.

At the present state of the art, unsupervised learning is not well understood and is still the subject of research. This research is currently of interest to the govemment because rnilitary situations often do not have a <lata set available to train a network until a conflict arises.

2.5.3 Learning **Rates**

The rate at which ANNs learn depends upon several controllable factors. In selecting the approach there are many trade-offs to consider. Obviously, a slower rate means a lot more time is spent in accomplishing the off-line learning to produce an adequately trained system. With the faster learning rates, however, the network may not be able to make the fine discriminations possible with a system that learns more slowly. Researchers are working on producing the best ofboth worlds.

Generally, several factors besides time have to be considered when discussing the off-line training task, which is oftendescribed as "tiresome." Network complexity, size, paradigm selection, architecture, type of learning rule Of rules employed, and desired accuracy must all be considered. These factors play a significant role in determining how long it will take to train a network. Changing any one of these factors may either extend the training time to an unreasonable length or even result in an unacceptable accuracy.

Most Iearning fiinctions have some provision for a learning rate, Of learning constant. Usually this terrn is positive and between z~ro and one. If the learning rate is greater than one, it is easy for the learıijp.g algorithm to ov:.~fshoot in correcting the weights, and the network will oscillate. Small values of the l~a:rning rate will not correct

the current error as quickly, but if small steps afe taken in correcting errors, there is a good chance of arriving at the best minimum convergcncc.

2.5.4 Learning Laws

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

Hebb's Rule: The first, and undoubtedly the best known, learning rule as introduced by Donald Hebb. The description appeared in his book *The Organization of Behavior* in 1949 [14]. His basic rule is: If a neuron receives an input from another neuron, and if both are highly active (mathematically have the same sign), the weight between the neurons should be strengthened.

Hopfield Law: It is similar to Hebb's rule with the exception that it specifies the magnitude of the strengthening Of weakening. It states, "If the desired output and the input are both active of both inactive, increment the connection weight by the learning rate, otherwise decrement the weight by the learning rate." [15].

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

The way that the İj)~lta Rule w;~r~s is that the delta error in the output layer is transformed by the derivative of the trans{Jf iunction and is then used in the previous neural layer to adjust input connection w~}ghts. In other words, this error is back-propagated into previous layers one layer at atime. The process of'back-propagating the network errors continues until the first layer is reached. The network type called

Feedforward, Back-propagation derives its name from this method of computing the error term.

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

The Gradient Descent Rule: This rule is similar to the Delta Rule in that the derivative of the transfer function is still used to modify the delta error before it is applied to the connection weights. Here, however, an additional proportional constant tied to the learning rate is appended to the final modifying factor acting upon the weight. This rule is commonly used, even though it converges to a point of stability very slowly. It has been shown that different learning rates for different layers of a network help the learning process converge faster. In these tests, the learning rates for those layers close to the output were set lower than :those layers near the input. This is especially important for applications where the input <lata is not derived from a strong underlying model.

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

Further, the size ofthe neighborhood can vary during the training period. The usual paradigm is to start with a larger definition of the neighborhood, and narrow in as the training process proceeds. Because the winning element is defined as the one that has the closest match to the input pattern, Kohonen networks model the distribution of the inputs. This is good for statistical or topological modeling of the data and is sometimes referred to as self-organizingmaps or self-organizingtopologies.

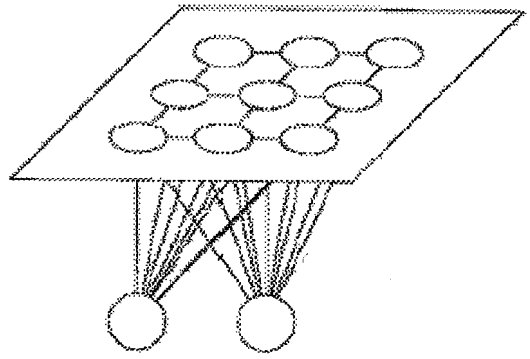
2.6 Advanced Neural Networks

Many advanced algorithms have bee:t\.invented since the first simple neural network. Some algorithms are based on the sanı~ assumptions or learning techniques as

the SLP and the MLP. A very different approach however was taken by Kohonen, in his research in self-organising networks.

2.6.1 Kohonen Self-Organislng Networks

The Kohonen self-organising networks have a two-layer topology. The first layer is the input layer, the second layer is itself a network in a plane. Every unit in the input layer is connected to all the nodes in the grid in the second layer. Furthermore the units in the grid function as the output nodes.



Input nodes

Figure 2.9 The Kohonen Topology

The nodes in the grid are only sparsely connected. Here each node has four immediate neighbours.

Algorithm

The network (the units in the grid) is initialised with small random values. A neighbourhood radius is set to a large value. The input is presented and the Euclidean distance between the input and each output node is calculated. The node with the minimum distance is selected, and this node, together with its neighbors within the neighbourhood radius, will have their weights modified to increase similarity to the ·lıput. The neighborhood radius decreases over time to let areas of the network be specialised-toa pattern.

The algorithm results in.~.n~t~QfWi"\\Y{l~fe groups of nodes respond to each class thus creating a map ofthe found.~lc,1.şşes.

The big difference in the ıı~~~ıig algorithm; q~m,paredwith the MLP, is that the Kohonen self-organising net usesunsupervised lea~in~- But after the learning period when the network has mapped the test patterns, it is 'iHe operaıors responsibility to label the different patterns accordingly.

2.6.2 Hopfleld Networks

The Hopfield network is a fully connected, symmetricallyweighted network where each node functions both as input and output node. The idea is that, depending on the weights, some states are unstable and the network will iterate a number of times to settle in a stable state.

The network is initialised to have a stable state with some known patterns. Then, the function of the network is to receive a noisy or unclassified pattern as input and produce the known, learnt pattern as output.

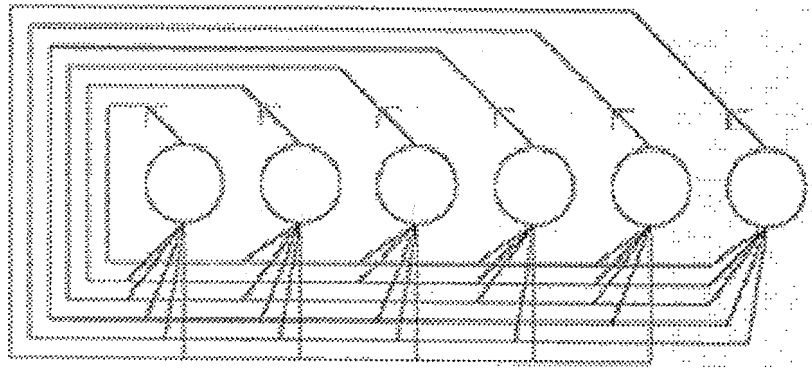
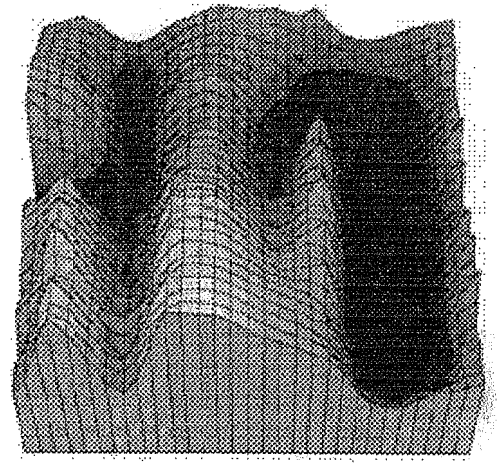


Figure 2.10. Hopfield Topology

Algorrthm

The energy flınction for the network is minimised for each of the patterns in the training set, by adjusting the connection weights. An unknown pattern is presented for the network. The network iterates until convergence. The Hopfield network can be visualised by means of the Energy Landscape, where the hollows represent the stored patterns. in the iterations of the Hopfield network the energy will be gradually minimiseduntil a steady state in one ofthe basins is reached.



Figüre 2.11. Energy Landscape

2.7 Problems using Neural Networks

2.7.1 Local Minimum

All the NN in this paper are described in their basic algorithnı. Several suggestions for improvements and modifications have been made. One of the well-known problems in the MLP is the *loca! minimum:* The network does not settle in one of the learned minima but instead in a local minimum in the Energy landscape

Approaches to avoid local minimum:

- The *gain term* in the weight adaption function can be lowered progressively as the network iterates. This would at first let the differences in weights and energy be large, and then hopefully when the network is approaching the right solution, the steps would be smaller. The tradeoff is when the gain term has decreased the network will take a longer time to converge to right solution.
- A local minimum can be caused by a bad internal representation of the patterns. This can be aided by the adding more internal nodes .to the network.
- An extra term can be added to the weight adaption: the *Momentum term.* The Momentum term should let the weight change be large if the current change in energy is Iarge.
- The network gradient descent can be disrupted by adding random noise to ensure sure the sytem will take unequal steps toward the solution. This solution has the advantage, that it requires no extra computation time.

A similar problem is known in the Hopfield Network as *metastable states.* That is when the network settles in a state that is not represented in the stored patterns. One way to minimise this is by adjusting the number of nodes in the network (N) to the number of patterns to store, so that the number of patterns does not exceed 0.15N. Another solution is to add a probabilistic update rule to the Hopfield network. This is known as the Boltzman machine.

2.7.2 Practical Problems

There are some practical pro~İ~jş.,applying neural networks to applications.

It is not possible to\~P,owin ad~~~~ the ideal network for an application, So every time a NN is to be built i'n an application, it r~~~irestests and experiments with different network settings or topologies to find a sotl;lti<;m that performs well on the given application. This is a problem because most NN requires a long training period - many iterations of the same pattern set. And even after many iterations there is no way other

that testing to see whether the network is efficiently mapping the training sets. A sohıtion for this might be to adapt newer NN technologies such as the bumptree which need only one run through the training set to adjust all weights in the network. The most commonly used network still seems to be the MLP and the RBF3 even though altematives exist that can drastically shorten processing time.

In general most NN include complex computation, which is time consuming. Some of these computations could gain efficiency if they were to be implemented on a parrallel processing system, but the hardware implementation raises new problems of physical limits and the NN need for changeability.

2.8 **Summary**

As we have discussed neural networks classified according to their learning processes into 'two types, supervised learning and unsupervised learning. Also this chapter discussed the various types of neural networks structures and algorithms. The most commonly used neural network configurations known as multilayer perceptron (MLP) are described. Other structures discussed in this chapter include recurrent (feedback) neural networks and radial hasis function (RBF) network. Also we have explained a brief description of Kohenon self-orginising networks and Hopfield networks. Finally we have discussed the problems using neural networks including the local minimumand practical problems.

CHAPTER THREE

NEURAL NETWORKS APPLICATIONS

3.1 Overview

This chapter presents a brief description of some artificial neural networks applications. The section below provides an understanding of how neural networks are currently being used and the researches area in artificial neural networks. The applications that artificial neural networks cover in this chapter such as language processing, character recognition, servo control and pattern recognition are described briefly. Also there will be a sufficient description about neural networks applications in image compression, and some applications area in medicine and business, also the applications in arts and telecommunications. Last section presents a determination if an application is a neural network candidate and how to determine it.

3.2 How Artificial Neural Networks Are Being Used

Artificial neural networks are undergolng the change that occurs when a concept leaves the academic environment and is thrown into the harsher world of users who simply want to get a job done. Many of the networks now being designed are statistically quite accurate but they still leave a bad taste with users who expect computers to solve their problems absolutely. These networks might be 85% to 90% accurate. Unfortunately, few applications tolerate that level of error.

While researchers continue to work on improving the accuracy of their "creations", seme explorers are finding uses for the current technology.

In reviewing this state of the art, it is hard not to be overcome by the bright promises or tainted by the unachieved realities. Currently, neural networks are not the user interface which translates spoken works into instructiopsJor a machine, but some day they will. Someday, VCRs, home secu1t~ systemsf ,l:>, players, and word processors will simply be activated by voice. 1'01.ıch screen anq;> voiç~ editing will replace the word processors of today while bringing spreadsheets and data bases to a level of usability pleasing to most everyone. But for now, neural networks ate simply

entering the marketplace in niches where their statistical accuracy is valuable as they await what will surely come.

Many of these niches indeed involve applications where answers are nebulous. Loan approval is one. Financial institutions make more money by having the lowest bad loan rate tfiey can achieve. Systems that are "90% accurate" might be an improvement over the current selection process. Indeed, some banks have proven that the failure rate on loans approved by neural networks is lower than those approved by some of their best traditional methods. Also, some credit card companies are using neural networks in their application screening process.

This newest method of seeking the future by analyzing past experiences has generated its own unique problems. One ofthose problems is to provide a reason behind the computer-generated answer, say as to why a particular loan application was denied. As mentioned throughout this report, the inner workings of neural networks are "black boxes." Some people have even called the use of neural networks "voodoo engineering." To explain how a network learned and why it recommends a particular decision has been difficult. To facilitate this process of justification, several neural network tool makers have provided programs which explain which input through which node dominates the decision making process. From that information, experts in the application should be able to infer the reason that a particular piece of <lata is important.

Besides this filling of niches, neural network work is progressing in other more promising application areas. The next section of this chapter goes through some of these areas and briefly details the current work. This is done to help stimulate within the reader the various possibilities where neural networks might o:tfer solutions, possibitities such as language processing, character recognition, image compression, pattern recognition among others.

3.3 **Language** Precessing

Language processing encompasses a•. w;i,de variety of applications. These applications include text-to-speeclı,IcÇQnversion;'itıl~füdinput för machines, automatic language translation, secure voice ~~)(~d Iocks, \aut~.m.~tic transcription, aids for the deaf aids för the physically disabled which respond io(vqJçe commands, and natura! language processing.

Many companies and universities are researchlng how a computer, via ANNs, could be programmed to respond to spoken commands. The potential economic rewards are a proverbial gold mine. If this capability could be shrunk to a chip, that chip could become part of almost any electronic device sold today. Literally hundreds of millions ofthese chips could be sold.

This magic-like capability needs to be able to understand the 50,000 mest commonly spoken words. Currently, according to the academic journals, mest of the hearing-capable neural networks are trained to only one talker. These one-talker, isolated-word recognizers can recognize a few hundred words. Within the context of speech, with pauses between each word, they can recognize up to 20,000 words.

Seme researchers are touting even greater capabilities, but due to the potential reward the true progress and methods involved, are being closely held. The most highly touted, and demonstrated, speech-parsing system comes from the Apple Corporation. This network, according to an April 1992 Wall Street Journal article, can recognize mest any person's speech through a limited vocabulary.

This works continues in Corporate America (particularly venture capital land), in the universities, and in Japan.

3.4 Character Recegnition

Character recognition is another area in which neural networks are providing solutions. Some of these solutions are beyond simply academic curiosities. HNC Inc., according to a HNC spokesman, markets a neural network based product that can recognize hand printed characters through a scanner. This product can take cards, like a credit card application form, and put those recognized characters into adata base. This product has been out fer two and a half years. It is 98% to 99% accurate for numbers, a little less for alphabetical characters. Currently, the system is built to highlight chasaeters below a certain percent probability of being right so that a user can manually fill in what the computer could not. This product is in use by banks, financial institutions, and credit card companies.

Odin Corp., according to a press release in the:1,l~;y,~p:ıber 4, 1991 Electronic Engineering Tirnes, has also proved capable of recpğrıi?J11g charaqter:s, ıı:ı,pluding cursive. This capability utilizes Odin's proprietary Quantum Neural Network sqf;tware

package called, QNspec. It has proven uncannily successful in analyzing reasonably good handwriting. It actually benefits from the cursive stroking.

The largest amount of research in the field of character recognition is airned at scanning oriental characters into a computer. Currently, these characters require four or five keystrokes each. This complicated process elongates the task of keying a page of text into hours of drudgery. Several vendors are saying they are elese to commercial products that can scan pages.

3.5 Pattern **Recognition**

Recently, a number of pattern recognition applications have been written about in the general press. The Wall Street Journal has featured a system that can detect bombs in luggage at airports by identifying, from small variances, patterns from within specialized sensor's outputs. Another article reported on how a physician had trained a back-propagation neural network on data collected in emergency rooms from people who felt that they were experiencing a heart attack to provide a probability of a real heart attack versus a false alarm. His system is touted as being a very good discriminator in an arena where priority decisions have to be made all the time.

Another application involves the grading of rare coins. Digitized images from an electronic camera are fed into a neural network. These images include several angles of the front and back. These images are then compared against known patterns which represent the various grades for a coin. This system has enabled a quick evaluation for about $15 as opposed to the standard three-person evaluation which costs $200. The results have shown that the neural network recommendations are as accurate as the people-intensive grading method.

Yet, by far the biggest use of neural networks as a recognizer of patterns is within the :field known as quality control. A number of automated quality applications are now in use. These applications are designed to :find that one in a hundred or one in a thousanıj, part that is defective. Human inspectors become fatigued or distracted. Systems now evaluate solder joints, welds, cuttings, and glue applications. One car manufacturer is now even prototyping a system which evaluates the color of paints. This system digitizes pictures of new, batclıesz,9{;.paint to determine if they are the right shades.

Another major area where neural networks are being built into pattern recognition systems is as processors for sensors. Sensors can provide so much <lata that the few meaningful pieces of information can become lost. People can lose interest as they stare at screens looking for "the needle in the haystack." Many of these sensor-processing applications exist within the defense industry. These neural network systems have been shown successful at recognizing targets. These sensor processors take data from cameras, sonar systems, seismic recorders, and infrared sensors. That data is then used to identify probable phenomenon.

Another field related to defense sensor processing is the recognition of patterns within the sensor <lata of the medical industry. A neural network is now being used in the scanning of PAP smears. This network is trying to do a better job at reading the smears than can the average lab technician. A missed diagnosis is a too common problem throughout this industry. In many cases, a professional must perceive patterns from noise, such as identifying a fracture from an X-ray or cancer from a X-ray "shadow." Neural networks prornise, particularly when faster hardware becomes available, help in many areas of the medical profession where data is hard to read.

3.6 **Servo** Control

Controlling complicated systems is one of the more promising areas of neural networks. Most conventional control systems model the operation of all the system's processes with one set of formulas. To customize a system fora specific process, those formulas must be manually tuned. It is an intensive process which involves the tweaking of parameters until a combination is found that produces the desired results. Neural networks offer two advantages. First, the statistical model of neural networks is more complex that a simple set of formulas, enabling it to handle a wider variety of operating conditions without having to be retuned. Second, because neural networks Iearn on their own, they don't require control system's experts, just simply enough historical <lata so that they can adequately train themselves.

Within the oil industry a neural network hao&,l>e.e.rı applied to the refinery process. The network controls the flow of mı:ıte.rials and is t§ı;ıte.ci. tp. do that in a more vigilant fashion than distractible humans.

NASA is working on a system to control the shuttle during in-flight maneuvers. This system is known as Martingale's Parametric Avalanche.Another prototype

application is k:nown as ALVINN, for Autonomous Land Vehicle in a Neural Network. This project has mounted a camera and a laser range finder on the roof of a vehicle which is being taught to stay in the middle ofa winding road.

British Columbia Hydroelectric funded a prototype network to control operations of a power-distribution substation that was so successful at optimizing four large synchronous condensors that it refused to let its supplier, Neural Systems, tak:e it out.

3.7 Image Compression

Computer images are extremely data intensive and hence require large amounts of memory for storage. As a result, the transmission of an image from one machine to another can be very time consuming. By using data compression techniques, it is possible to remove some of the redundant information contained in images, requiring less storage space and less time to transmit. Neural networks can be used for the purpose of image compression.

Neural network archltecture suitable for solving the image compression problem is shown below. This type of structure--a large input layer feeding into a small hidden layer, whlch then feeds into a large output layer, is referred to as a bottleneck type network. The idea is this: suppose that the neural net shown below had been trained to implement the identity map. Then, a tiny image presented to the network as input would appear exactly the same at the output layer.

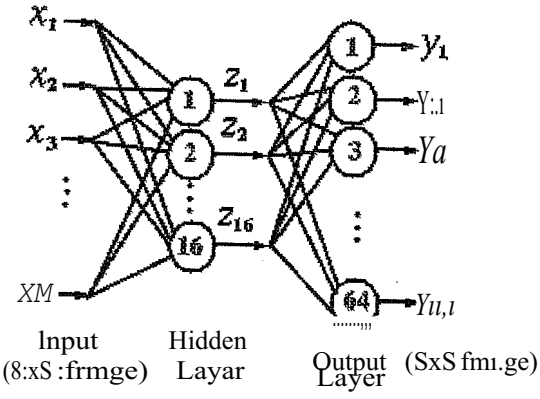


Figure 3.1. Bottleneck-type Neural Net Architecture for Image Compression

In this case, the network could be used for image compr~ş.ş!cın by breaking it in two as shown in the Figure below. The transmitter encodes and the*transmits the output ofthe hidden layer (only 16 values as compared to the 64 values of the original image).The receiver receives and decodes the 16 hidden outputs and generates the 64 outputs. Since

the network is implementing an identity map, the output at the receiver is an exact reconstruction of the original image.

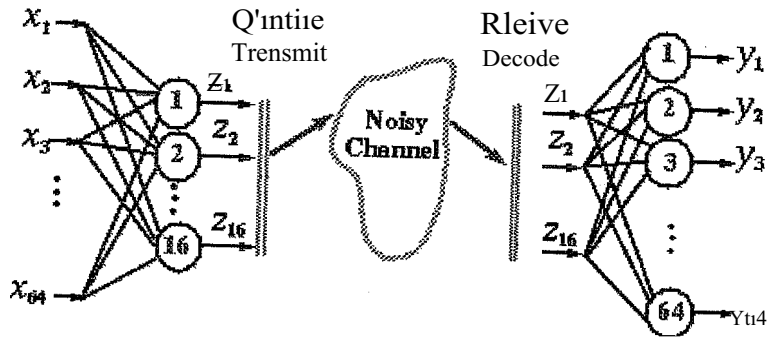


Figure 3.2. The Image Compression Scheme using the Trained Neural Net

Actually, even though the bottleneck takes us from 64 nodes down to 16 nodes, no real compression has occurred because unlike the 64 original inputs which are 8-bit pixel values, the outputs of the hidden layer are real-valued (between -1 and 1), which requires possibly an infinite number of bits to transmit. True image compression occurs when the hidden layer outputs are quantized before transmission. The Figure below shows a typical quantization scheme using 3 bits to encode each input. In this case, there are 8 possible binary codes which may be formed: 000, 001, 010, 011, 100, 101, 110, 111. Each of these codes represents a range ofvalues fora hidden unit output. For example, consider the first hidden output. When the value ofis between -1.0 and -0.75, then the code 000 is transmitted; when is between 0.25 and 0.5, then 101 is transmitted. To compute the amount ofimage compression (measured in bits-per-pixel) for this level of quantization, we compute the ratio of the total number of bits transmitted: to the total number of pixels in the original image: 64; so in this case, the compression rate is given as bits/pixel. Using 8 bit quantization of the hidden units gives a compression rate of bits/pixel.

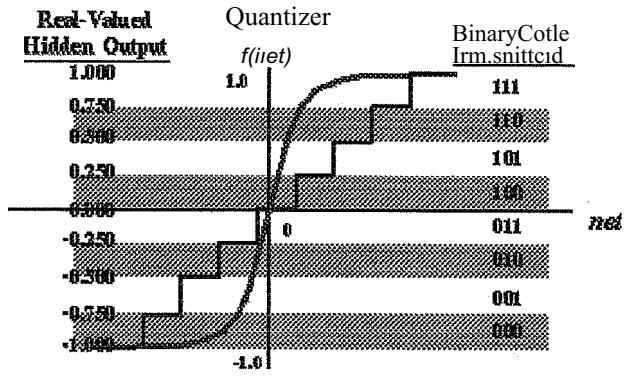


Figure 3.3. The Quantization ofHidden Unit Outputs

The training of the neural net proceeds as follows, a 256x256 training image is used to train the bottleneck type network to learn the required identity map. Training input-output pairs are produced from the training image by extracting small 8:x.8 chunks of the image chosen at a uniformly random Iocation in the image. The easiest way to extract such a random chunk i s to generate a pair of random integers to serve as the upper left hand corner of the extracted chunk. In this case, we choose random integers *i* and *j,* each between 0 and 248, and then *(ij)* is the coordinate of the upper left hand corner of the extracted chunk. The pixel values of the extracted image chunk are sent (left to right, top to bottom) through the pixel-to-real mapping shown in the Figure below to construct the 64-dimensional neural net input, Since the goal is to learn the identity map, the desired target for the constructed input is itself hence, the training pair is used to update the weights ofthe network.

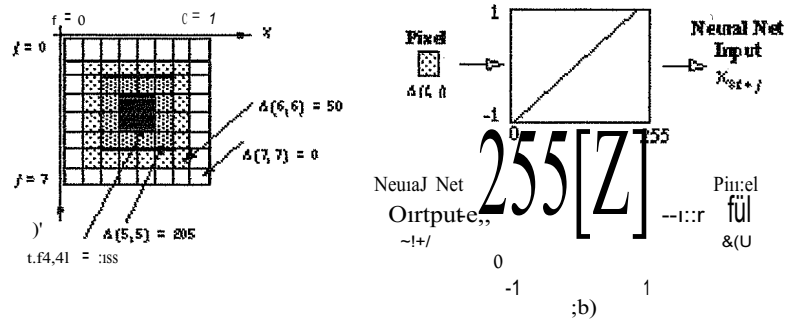


Figure 3.4. The Pixel-to-Real and Real-to-Pixel Conversions

ünce training is complete, image cqnıpression is demonstrated in the recall phase. In this case, we still present ıtıe neuraln~f with 8x8 chunks of the image, but now instead of randomly selecting the location of each chunk, we select the chunks in sequence from left to right and from top to bottom, .For each such 8x8 chunk, the output

the network can be computed and displayed on the scfeen to visually observe the performance of neural net image compression. In addition, the 16 outputs of the hidden layer can be grouped into a 4x4 "compressed image", which can be displayed as well.

3.8 Neural Networks in Business

Business is a diverted field with several general areas of specialisation such as accounting Of financial analysis. Almost any neural network applicaiion would fit into one business area Of 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 IOf 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.

3.8.1 Marketing

There is a marketing application which has been integrated with a neural network system. The Alrline 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. Additionally, the application's environment changed rapidly and constantly, which required a continuously adaptive solution. The system is used to monitor and recommend book:ing 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. [Hutchison & Stephens, 1987J[20].

While it is significant that;l1~B:tal netwq~~~,have befn applied to this problem, it is also important to see that thistintelligenf t~c:hnolo~k~~rı be integrated with expert systems and other approaches tq. make a functional syst~ıu.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.

3.8.2 Credlt Evaluation

The HNC Company, founded by Robert Hecht-Nielsen [21], has developed several neural network applications. üne of them is the Credit Scoring system which increases 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 underwriting system was developed by the Nestor Company. This system was trained with 5048 applications of which 2597 were certified. The <lata related to property and borrower qualifications. In a conservative mode the system agreed on the underwriters 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.

Loan granting is one area in which neural networks can aid humans, as it is an area not based on a predetermined and preweighted criteria, but answers are instead nebulous. Banks want to make as much money as they can, and one way to do this is to lower the failure rate by using neural networks to decide whether the bank should approve the loan. Neural networks are particularly useful in this area since no process will guarantee 100% accuracy. Even 85-90% accuracy would be an improvement over the methods humans use.

In fact, in some banks, the failure rate of loans approved using neural networks is lower than that of some of their best traditional methods. Some credit card companies are now beginning to use neural networks in deciding whether to grant an application.

The process works by analyzing past failures and making current decisions based upon past experience. Nonetheless, this creates its own problems. For example, the bank or credit company must justify their decision to the applicant. The reason "my neural network computer recommended against it" simply isn't enough for people to accept. The process of explaining how the network learned and on what characteristics the neural network made its decision is difficult. As we alluded to earlier in the history of neural networks, self-modifying code is very di:fficult to debug and thus difficult to trace. Recording the steps it went througb isn't enough, as/ it might be using conventional computing, because even the individual steps the nı:lural network went

through have to be analyzed by humarı beings, or possibly the network itself to determine that a particular piece of data was crucial in the decision-making process.

3.9 Nenral Networks in Medicine

Artificial Neural Networks are currently a 'hot' research area in medicine and it is believed that they will receive extensive application to biomedical systems in the next few years. At the moment, the research is mostly on modelling parts of the humarı body and recognising diseases from various seans (e.g. cardiograms, CAT seans, ultrasonic seans, etc.).

Neural networks are ideal in recognising diseases using seans since there is no need to provide a specific algorithm on how to identify the disease. Neural networks learn by example so the details of how to recognise the disease are not needed. What is needed is a set of examples that are representative of all the variations of the disease. The quantity of examples is not as important as the 'quantity'. The examples need to be selected very carefully ifthe system is to perform reliably and efficiently.

3.9.1 Modeling and Diagnosing the Cardiovascular System

Neural Networks are used experimentally to model the humarı cardiovascular system. Diagnosis can be achieved by building a model of the cardiovascular system of an individual and comparing it with the real time physiological measurements taken from the patient. If this routine is carried out regularly, potential harmful medical conditions can be detected at an early stage and thus make the process of combating the disease much easier.

A model of an individual's cardiovascular system must mimic the relationship among physiological variables (i.e., heart rate, systolic and diastolic blood pressures, and breathing rate) at different physical activity levels. If a model is adapted to an individual, then it becomes a model of the physical condition of that individual. The simulator will have to be able to adapt to the features of any individual without the supervlsion of an expert. This calls for a neuralg~twork.

Another reason that justifies\;h~ use of AJ):~lNLtechnology is the ability of ANNs to provide sensor fusion which is tfüf eombining of valu~ş%j:i,:pm several different sensors. Sensor fusion enables the ANNs to leam complex relatiqnships among the individual sensor values, which would otherwise be lost if the values were individually analysed. In medical modelling and diagnosis, this implies that even though each sensor in a set

may be sensitive only to a specific physiological variable, ANNs are capable of detecting complex medical conditions by fusing the <lata from the individual biomedical sensors.

3.9.2 Electronic Noses

The two main components of an electronic nose are the sensing system and the automated pattem recognition system. The sensing system can be an array of several different sensing elements (e.g., chemical sensors), where each element measures a different property of the sensed chemical, Of it can be a single sensing device (e.g., spectrometer) that produces an array of measurements for each chemical, or it can be a combination, Each chemical vapor presented to the sensor array produces a signature Of pattem characteristic of the vapor. By presenting many different chemicals to the sensor array, a database of signatures is built up. This database of labeled signatures is used to train the pattern recognition system. The goal of this training process is to configure the recognition system to produce unique classifications of each chemical so that an automated identification can be implemented.

The quantity and complexity of the <lata collected by sensors array can make conventional chemical analysis of <lata in an automated fashion difficult. üne approach to chemical vapor identification is to build an array of sensors, where each sensor in the array is designed to respond to a specific chemical. With this approach, the number of unique sensors must be at least as great as the number of chemicals being monitored. It is both expensive and difficult to build highly selective chemical sensors.

Artificial neural networks (ANNs), which have been used to analyze complex <lata and to recognize pattems, are showing promising results in chemical vapor recognition. When an ANN is combined with a sensor array, the number of detectable chemicals is generally greater than the number of sensors [22]. Also, less selective sensors which are generally less expensive can be used with this approach. ünce the ANN is trained for chemical vapor recognition, operation consists of propagating the sensor data through the network. Since this is simply a series of vector-matrix multiplications, unknown chemicals can be rapidly.identifiedin the field.

Electronic noses that incorporate ~s. have bee'ı:l;t,,;demonstrated in vanous applications. Some of these applications wil1~~:qi§.5uss~Pat;rin th.~,,%aper. Many ANN configurations and training algorithms have been used to bul~;;eelectronic noses including backpropagation-trained, feed-forward networks; fuzzy AR.Tmaps;Kohonen's

self-organizing maps (SOMs); learning vector quantizers (LVQs); Hamming (~~two~JI~ı Boltzmann machines; and Hopfield network:s. Figure 3.5 iilustrates the basic \~~~.~tic 'of an electronic nose.

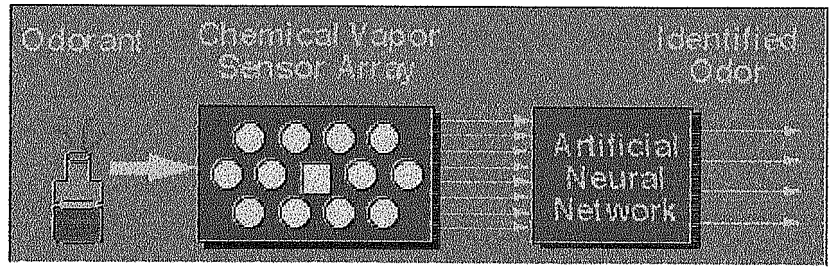


Figure 3.5. Schematic diagram of an electronic nose

Because the sense of smell is an important sense to the physician, an electronic nose has applicability as a diagnostic tool. An electronic nose can examine odors from $he body (e.g., breath, wounds, body fluids, ete.) and identify possible problems. Odors in the breath can be indicative of gastrointestinal problems, sinus problems, infections, diabetes, and liver problems. Infected wounds and tissues emit distinctive odors that can be detected by an electronic nose. Odors coming from body fluids can indicate liver and bladder problems. Currently, an electronic nose for examining wound infections is being tested at South Manchester University Hospital [23].

A more futuristic application of electronic noses has been recently proposed for telesurgery [24]. While the inclusion of visual, aural, and tactile senses into telepresent systems is widespread, the sense of smell has been largely ignored. An electronic nose will potentially be a key component in an olfactory input to telepresent virtual reality systems including telesurgery. The electronic nose would identify odors in the remote surgical environment. These identified odors would then be electronically transmitted to another site where an odor generation system would recreate them.

3.9.3 Instant Physician

An application developed in the mid- 1980s called the "instant physician" trained an autoassociative memory neural network to store a large number of medical records, each of which includes information on symptoms, diagnosis, and treatment for a particular case. A:fter training, the net can be presented withjnput consisting ofa set of symptoms; it will then find the full stored pc1,ttyrn that represents, t~y "best" diagrıosis and treatment.

3.9.ft Medical Image Analysis

ANNs are used in the analysis of medical images from a variety of imaging modalities. Applications in this area include tumor detection in ultra-sonograms, detection and classification of microcalcifications in mammograms, classification of chest x-rays, tissue and vessel classification in magnetic resonance images (MRI), x-ray sp\ectral reconstruction, determination of skeletal age from x-ray images, and determinaiion of brain maturation. At Pacific Northwest National Laboratory [25], ~Ns are being developed to examine thallium scintigram images of the heart and identify the existence of infarctions. Another project at Pacific Northwest National La~oratory uses ANN technology f0 aid in the visualization of three-dimensional ultrasonic images.

3.,.5 Medical Diagnostic Aldes

The application of ANNs in diagnosing heart attacks received publicity in the Wall St;~et Joumal when the ANN was able to diagnose with better accuracy than phyşicians. This application is signiflcant because it was used in the emergency room where the physicians are not able to handle large amounts of data.

A commercial product employs ANN technology in the diagnosis of cervical cancer by examining pap smears. In clinical use, this product has proven to be superior over.human diagnosis ofpap smears.

In the United Kingdom, an ANN used in the early diagnosis of myocardial infarction is currently undergoing clinical testing at four hospitals. At the research level, *ANNs* are used in diagnosing ailments such as heart murmur, coronary artery disease, lung disease, and epilepsy.

This technology is also being used in the interpretation of electrocardiograms (ECG) and electroencephalograms (EEG).

S.Ul Applicatiens in the Arts

We now turn to the artistle uses of NNs. Currently, this is a wide-open fıeld; exploration has just begun in most cases, and we've barely scratched the surface of possibilities. The ideas below are mostly specul~tions on what networks could do, the sorts of tasks they could be applied to in the arts, semetimes based on applications that have already been done in scientiflc or engineering domains, and sometimes just based on imaginative speculation. As such, these ideas are intended to spark people's

imaginations further in the search for innovative uses of this powerful and fle:xible new technology.

The main place where neural networks have been put to creative and artistic use so far is in *music,* as witnessed by the recent publication of the book, Music and Connectionism (Todd and Loy, 1991) [26]. Several applications have been done in this area, ranging from psychological models of human pitch, chord, and melody perception, to networks for algorithmic composition and performance control. Generally speaking, the applications here (and in other fields) can be divided into two classes: "input" and "output". The input side includes networks for recognition and understanding of a provided stimulus, for instance speech recognition, or modelling how humans listen to and process a melody. Such applications are useful for communication from human to machine, and for artistic analysis (e.g. musicologically, historically) ofa set of inputs. The output side includes the production of novel works, applications such as music composition or drawing generation. "Input" tasks tend to be much more difficult than "output" tasks (compare the state-of-the-art in speech recognition versus speech production by computers), so most of the network applications so far have focussed on creation and generation of output, but continuing research has begun to address this imbalance.

On the "input" side in musical applications, Sano and Jenkins (1991) [26] have modelled human pitch perception; Bharucha (1991) [26] (and others) have modelled the perception and processing of harmony-and chords; Gjerdingen (1991) [26] has explored networks that understand more complex musical patterns; and Desain and Honing (1991) have devised a network for looking at the quantization of musical time and rhythm. Dolson (1991) [26] has also suggested some approaches to musical signal processing by neural networks, including instrument recognition, generation, and modification. In this regard, musical applications of networks have much to gain from the vast literature on networks for speech processing (primarily recognition=see Lippmann, 1989) [27].

On the "output" side, several network models of music composition have been devised. Todd (1991a) and M~eer (199.l)}il~Bluse essentially the dynamic sequential network approach mentioned e~l~er, in whi.JH>a.net~ork is trained to map from one time-chunk of a piece of music to the following tim~~~i~P-k (e.g. measure N as input should produce measure N+1 as output). The network's outpıits are then connected back to its inputs for the creation phase, and a new measure 1 is provided to begin the

network down a new dynamic patlı, creating one measure after another, and all the while incorporating the sorts of features it leamed from its training examples. In this way, new pieces that have a sound like Bach Of Joplin (ofa combination of both!) can be created, if the network is first trained on these composers. But the problems mentioned earlier of lack of higher-level structure emerge, and these compositions tend towander, having no clear direction, and seldom ending up anywhere in particular. Approaches for Iearning and using hierarchical structure are being devised, and Lewis ,;(1991a) [26] describes one such method, in which the inputs to a network, rather than .the weights in the network, are modified during a learning stage, to produce an input \ which has a specified form Of character. Kohonen present still another method of .compoaition, which uses a network-style approach to build up a context-free grammar that models the music examples it's trained on.

Networks can also be used to generate musical performance parameters and ,tnstructions, as Sayegh (1991) [26] demonstrates in his paper ona network method for choosing correct chord fingering for a simple melody. Many other musical performance applications are possible, from synthesizer control to automatic rhythmic accompaniment generators; Todd (1991b) discusses some of these possibilities along with further ideas for musical applications of neural networks,

3.,11 Neural Networks in Telecommunications

The IEEE Communications Society is active in developing a list of state-of-the-art topics in communicatiorıs. Some of these are areas in which neural networks have a rôle,: such as signal processing for beam forming, adaptive antennas, consumer commıınications, radio resource management and mobility management.

Beam forming employs signal processing in transmitting information over multiple antennas. It is also used for receivers to create steerable arrays. The purpose of beamforming is to minimize interference whether this is caused by fading, reflections Of the effects of multi-user interference. If the channel is unknown Of is changing, an adaptive antenna system will prove to be an advantage. Adaptive antennas can also offer capacity enhancements Of allow higher bit-rates to be used.

Consumer products will soon have the capability of high-speed communications. This requires low cost and low power electronics. However, the domestic environment may not be RF-friendly so that an intelligent and adaptive receiver can improve the throughput without requiring an increase in transmitter power. One such wireless

communications standard is Bluetooth; Bluetooth has to compete with IrDA (Infrared Data Association) which is a line-of-sight system, whereas Bluetooth is not.

Wireless systems are demanding hlgher spectrum efficiency as applications become more bandwidth-hungry. Radio resource management is essential and requires dynamic channel assignment, interference avoidance, propagation prediction and automated planning techniques which are conventional neural network applications. Handoff requires a decisi on which is similar to a fuzzy Iogic rule.

When a user moves between a fixed and mobile platform, it will be essential that this user can enjoy the same services and applications transparently. Research continues into intelligent systems to implement dynamic routing, wireless ATM and location prediction.

3.12 How to Determine ifan Appheation is a Neural Network Candidate

As seen by the sections above, neural networks are being successfullyappliefl<faa number of areas. Each of these applications can be grouped into two broad categories. These categories offer a test for anyone who is considering using neural rietworks. Basically, a potential application should be examined for the following two criteria:

1. Can a neural network replace existing technologies in an area where small improvements in performance can result in a major economic impact? Examples of applications which meet these criteria are:
   - Loan approvals
   - Credit card approvals
   - Financial market predictions
   - Potential customer analysis for the creation of mailing lists
2. Can a neural network be used in an area where current technologies have proven inadequate to making a system viable? Examples of applications which meet these criteria are:
   - Speech recognition
   - Text recognition
   - Tai:getanalysis

(Another example where other technologies failed was in explosive detection at airports. Previous systems could not achleve the FAA mandated level of performance,

but by adding a neural network the system not only exceeded the performance, it allowed the replacement ofa $200,000 component.)

The most successful applications have been focused on a single problem in a high value, high volume, or a strategically important application.

The easiest implementation of neural networks occurs in solution where they can be made to be "plug compatible" with existing systems. To simply replace an existing element of a system with a neural network eases an installation. It also increases the likelihood of success. These "plug cornpaıible" solutions might be at the front end of many systems where neural networks can recognize patterns and classify data.

3.13 Summary

This chapter demonstrates applications of artificial neural networks in various fields. We have described briefly neural networks applications in Ianguage processing, character and pattern recognition, and servo control application. Also we have discussed the neural networks application in image compression and application fields in medicine and business includes some examples, in addition to applications in arts and telecommunication. Finally we have discussed how to determine if the application is a neural network candidate.

CHAPTER FOUR

NEURAL NETWORKS iN FRAUD DETECTION

4.1 Overview

Fraud detection is an important application of neural networks in the prediction techniques. This chapter shows the concept of fraud detection and how neural networks obtained high fraud coverage to detect credit card operations. Also this chapter concentrates on business application of neural networks through credit card fraud and fraud detection of credit card using neural networks. And finally this chapter discusses unsupervised learning and their applications to fraud detection.

4.2 Fraud Detection

In the fight against fraud, actions fal! under two broad categories: fraud prevention and fraud detection. Fraud prevention describes measures to stop fraud occurring in the first place. These include PINs for bankcards, Internet security systems for credit card transactions and passwords on telephone bank accounts. In contrast, fraud detection involves identifying fraud as quickly as possible once it has been perpetrated. We apply fraud detection once fraud prevention has failed, using detection methods continuously, as we will usually be unaware that fraud prevention has failed. In this article we are concerned solely with fraud detection.

Fraud detection must evolve continuously. ünce criminals realise that a certain mode of fraudulent behaviour can be detected, they will adapt their strategies and try others. Of course, new criminals are also attempting to commit fraud and many of these will not be aware of the fraud detection methods that have been successful in the past, and will adopt strategies that lead to identifiable frauds. This means that the earlier detection tools need to be applied as well as the latest developments.

Statistical fraud detection methods may be:':s,upervised' or 'unsupervised'. In supervised methods, mode1s are traitı.~Sl to discrimiaate between fraudulent and non-fraudulent behaviour, so that new observations can be assigned to classes so as to optimise some measure of classification performance. Of course, this requires one to be confident about the true classes of the original data used to build the mode1s;

uncertainty is introduced when legitimate transactions are mistakenly reported as fraud or when fraudulent observations are not identified as such. Supervised methods require that we have examples of both classes, and they can only be used to detect frauds of a type that have previously occurred. These methods also suffer from the problem of unbalanced class sizes: in fraud detection problems, the legitimate transactions generally far outnumber the fraudulent ones and this imbalance can cause misspecification of models. Brause et al (1999) [28] say that, in their database of credit card transactions, 'the probability offraud is very low (0.2%) and has been lowered in a preprocessing step by a conventional fraud detecting system down to 0.1%.' Hassibi (2000) remarks 'Out of some 12 billion transactions made annually, approximately 10 million - or one out of every 1200 transactions - turn out to be fraıidulent.'

In contrast, unsupervised methods simply seek those accounts, customers, ete. whose behaviour is 'unusual'. We model a baseline distribution that represents normal behaviour and then attempt to detect observations that show greatest departure from this norm. These can then be examined more closely. Outliers are a basic form of nonstandard observation that can be used for fraud detection.

This leads us to note the fundamental point that we can seldom be certain, by statistical analysis alone, that a fraud has been perpetrated. Rather, the analysis should be regarded as alerting us to the fact that an observation is anomalous, or more likely to be fraudulent than others - so that it can then be investigated in more detail. One can think of the objective of the statistical analysis as being to return a suspicion score (where we will regard a higher score as more suspicious than a lower one). The higher the score is, then the more unusual is the observation, or the more like previously fraudulent values it is. The fact that there are many different ways in which fraud can be perpetrated, and many different scenarios in which it can occur, means that there are many different ways of computing suspicion scores.

We can compute suspicion scores for each account in the database, and these scores can be updated as time progresses. By ordering accounts according to their suspicion score, we can focus attention on those with the highest scores, or on those that exhibit a sudden increase in suspicion score. If we have a limited budget, so that we can only afford to investigate a certain number of accounts or records, we can concentrate lnvestigation on those thought to be most likely to be fraudulent.

4.3 Credit Card Fraud

Credit card fraud is perpetrated in various ways but can be broadly categorised as application, 'missing in post', stolen/lost card, counterfeit card and 'cardholder not present' fraud. Application fraud arises when individuals obtain new credit cards from issuing companies using false personal information; applicatton fraud totaled fl0.2 million in 2000 (Source: APACS) and is the only type of fraud that actually declined between 1999 and 2000. 'Missing in post' (fl7.3m in 2000) describes the interception of credit cards in the post by fraudsters before they reach the cardholder. Stolen Of lost cards accounted for .:f98.9 million in fraud in 2000, but the greatest percentage increases between 1999 and 2000 were in counterfeit card fraud (.:f50.3m to .:fl02.8m) and 'cardholder not present' (i.e. postal, phone, intemet transactions) fraud (f29.3m to f56.8m). To commit these last two types of fraud it is necessary to obtain the details of the card without the cardholder's knowledge. This is done in various ways, including employees using an unauthorised 'swiper' that downloads the encoded information onto a laptop computer and hackers obtaining credit card details by intrusion into companies' computer networks. A counterfeit card is then made, Of the card details simply used for phone, postal Of Internet transactions.

Supervised methods to detect fraudulent transactions can be used to discriminate between those accounts Of transactions known to be fraudulent and those known (Of at least presumed) to be legitimate, For example, traditional credit scorecards (Hand and Henley, 1997) (30] are used to detect customers who are likely to default, and the reasons for this may include fraud. Such scorecards are based on the details given on the application forms, and perhaps also on other details, such as bureau information. Classification techniques, such as statistical discriminant analysis and neural networks, can be used to discriminate between fraudulent and non-fraudulent transactions to give transactions a suspicion score.

However, information about fraudulent transactions may not be available and in these cases we apply unsupervised methods to attempt to detect fraud. These methods are scarce in the literature and are less populaı;thıµı supervisE:d.m.E:thodsin practice as suspicion scores reflect a propensity to act anonı.ı:ı.fç:ı..ışly when compafE:g with previous behaviour. This is different to suspicion scores obtaiu~d. using supervis~<i techniques, which are guided to reflect a propensity to commit fraud in a manner already previously discovered. The idea behind suspicion scores from unsupervised methods is that

unusual behaviour or transactions can often be indicators of fraud. An advantage of using unsupervised methods over supervised methods is that previously undiscovered types of fraud may be detected. Supervised methods are only trained to discriminate between legitimate transactions and previously known fraud.

4.4 Neural Fraud Detection in Credit Card Operations

Pattern recognition is certainly one of the most relevant areas of application of neural networks. The range of concrete problems that may fall under this category is very wide. Probably one of the main sources of applications arises from "physical patterns". By this it means those coming from all sorts of signals, acoustic, graphical, or others. But another area of great practical interest and active research is the classification of what may be called social or economical patterns.

A clear instance are the long and widely used systems for credit scoring; that have essentially to decide whether or not a given petition is credit worthy or, instead, should be rejected.

The pattern, here, is a set of characteristics, financial, job related, familiar or other, of the credit applying person. This problem model is extendible to many other interesting instances as insurance policy decisions, :financial ratings, acceptance or rejection of credit cards operations.

This last problem has, of course, received a great deal of attention, under many different approaches, some of them Neural Networks based [31, 32]. There are even several commercial fraud detection systems with large neural components [33, 34]. In any case, it certainly has some peculiar characteristics derived from the different usage that such cards can have, though, we are going to concentrate only on the detection of possibly fraudulent operations. By this, we mean the usage of a given card that "may have been lost, stolen or falsified by an unauthorized person against the will of "its true owner. Certainly this possibility has to be taken into account even if the card's main use is as a credit device. But in some countries, most cards, rather than being used to finance purchases, simply provide a certain deferment of the payments due. Fraud is then the param~µrı,t risk issue,

Credit card fraud detectiôn also has two other highly peculiar characteristics. The first one is obviously the very limited time span in which the acceptance or rejection decision has to be made, The second one is the huge amount of credit card operations

that have to be processed ata given time. To just give a medium size example, in Spain more than 1.2 million Visa card operations take place in a given day, 98% of them being handled on line. Of course, just very few will be fraudulent (otherwise, the entire industry would have soon ended up being out of businesses), but this just means that the haystack where these needles are to be föund is simply enormous.

When considering the information to be used to rate a given operation, two distinct possibilities arise. In the first one, that we may call by-owner, operations are rated according to the usage history of the card owner. This approach requires the ability to fetch in a very fast way the pertinent owner's information from the usually huge databases of all card holder's historical records. We have to clarify what is meant for fast in this setting. Fraud prevention is very important to card issuers, but not to the point of making impractical or simply inconvenient the daily card utilisation by hundreds of thousands customers. When all the time needed for the remote connections and the basic operation processing is taken into account, a fraud detection system usually has no more than a rather small fractlon of a second to perform its task. This allotted time most likely will not be enough for large database queries. Of course, this may be alleviated if specially configured and dedicated hardware is available, but it will certainly result in higher start up and maintenance costs.

In any case, it is also true that such systems have additional advantages, being the most relevant one the deep and powerfül analysis that the past history of a customer allows when rating a certain operation. The specified systems may very well work in a sort of "deferred on-line" mode: even ifa given operation has to be authorised before an eventually negative rating has been completed, further incoming operations of its card will be effectively blocked. Notice that fraud may be fought over individual transactions, but it is won on the sum ofthem. Moreover, given the natıre.öf stolen or falsified card users, they will shift their attention from issuers with glöbally effective prevention systems to others less prepared.

There are situations, however, where a "by owner" systemis simply impossible to set up. This is case when a detection system is to beinstalled not at an issuer but rather in an "operation hub". That is, a central operation processing centre receives transactions from many sellers, distributes them to each particular card issuer, and relays its answer back to the originating sales point, The most important activity of such a hub is to streamline credit card trafik back and forth between sellers and issuers: therefore, it essentially does not have any owner information. Thus, contrary to what

happens for instance to card issuing institutions, the above hubs are not good candidates for a "by-owner" system. An altemative solution is to build a scoring system "by-operation", that is, a system that only uses the information of the operation itself When complemented with the previous operation history of the associated cards over a period of time, variables such as operation frequency, accumulated amounts, acceptance or rejection rates of previous operations, and so on, can be derived. These variables are obviously of great interest when deciding whether the current operation is legal or not.

Certainly, this Information is not as complete as the one that could be used in a "by-owner" system, but this approach has some clear strong points.

• Since only operations of an immediately previous peri od of time are considered, data querying is done on relatively small data bases, and operation scoring and authorisation can be performed in real-time, without requiring deferred processing.

• The small database size make possible to install such a system without having to use large dedicated hardware, thus reducing sensibly its costs.

• Its placement on a operation hub allows it to simultaneously service several issuers, without having to install individual systems at each of them.

The model for such a system is represented in the following diagram:

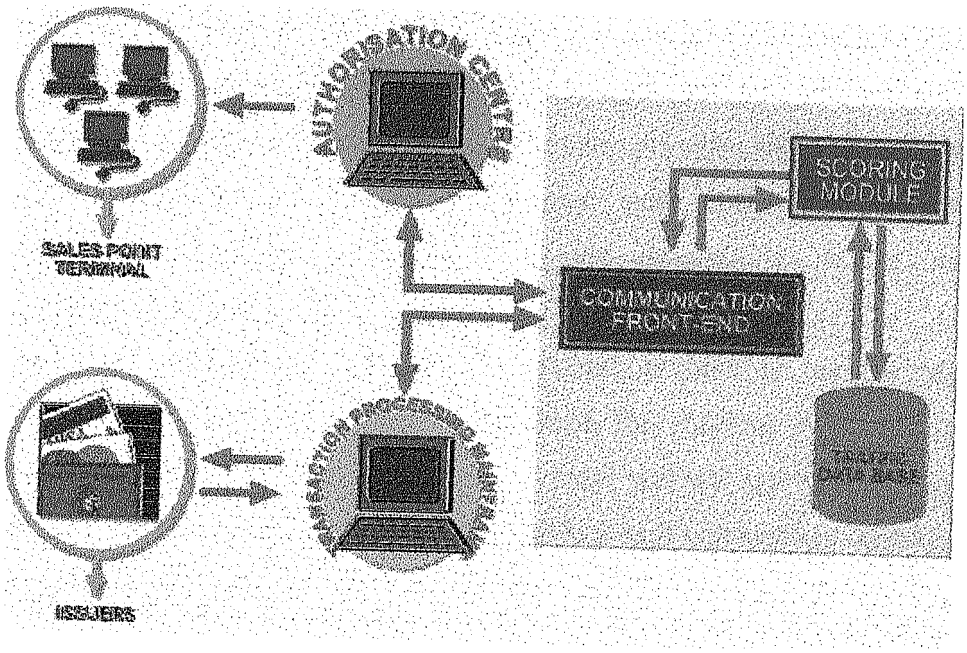


Figure 4.1. Fraud Detection Systern

Fraud deıection in credit card operation falls neatly in principle within the scope of pattem recognition procedures, and its solution can be sought through the construction of appropriate classifier functions. However, and has it may also be expected, it has

certain characteristics of its own that mak:e it a rather di:fficult problem. The most important is the great imbalance between good operations and fraudulent ones. This is not surprising at all: after all, card issuers intend to make a profit out of credit card use, and fraud directly diminishes that profit. Thus, they will already have implemented a number of fraud prevention methods. in one words, new fraud detection tools, neural Of other, are weapons to be used in a war already being fought.

From the point of view of the construction of neural detectors, this imbalance will certainly mak:e model training rather di:fficult. In fact, typical fraud rates may well be in the one per tens of thousands. This simply means that prior to the model construction, some kind of data segmentation has to be applied in order to lower these rates in the data sets to be used. Segmentation criteria have to be defined after a thorough statistical analysis of legal and fraudulent traffic among, for instance, the different geographical and sector areas for which independent detection modules are to be built.

If properly working, neural real time fraud detection systems will not only be technically feasible, but highly interesting from a purely economical point of view. However, their development has to overcome certain hurdles. First, rather extensive data analysis has to be performed on traffic information to obtain a meaningful set of detection variables. This analysis is also necessary to effectively segment that data in such a way that enormous imbalances between legal and fraudulent traffic do not overwhelm the later to the point of making detection possible. Moreover, di:fficult prior probabilities estimation and class overlapping may make ordinary multilayer perceptrons training essentially useless; it is thus necessary to devise new model building approaches ([35]).

Finally, we mention that although it is certainly possible to use the ratings of a neural credit card detection system as the hasis for absolute automated operation acceptance Of rejection criteria, an alternative, more realistic approach in by-operation systems is to use those ratings. If used jointly with a card's immediate operation history to make referral decisions, the performance and reliability of the system will be greatly enhanced. We should keep in mind th~){~~ negative c~~.şequences of a wrong denial of service decision. Even if this makes necesş;cy to mak:e the system work jointly with an Authorization Cemre (that many institutions will have in plaee anyway) this approach will still attack a significant portion of attempted fraud while making very good sense from a cost effectiveness point ofview.

4.5 Unsupervised Metbods and Their Application to Fraud Detection

As we mentioned above, the emphasis on fraud detection methodology is with supervised techniques. In particular, neural networks have proved popular - predictably, perhaps, given the attention they have received. Researchers who have used neural networks for supervised credit card fraud detection include Ghosh and Reilly (1994) [32], Aleskerov et al (1997) [36], Dorronsoro et al. (1997) [35], and Brause et al (1999) 28). However, unsupervised credit card fraud detection has not received attention in the literature.

Unsupervised fraud detection methods have been researched in the detection of computer intrusion (hacking), Here profiles are trained on the combinations of commands that a user uses most frequently in their account. If a hacker gains illegal access to the account then their intrusion is detected by the presence of sequences of commands that are not in the profile of commands typed by the legitimate user. Qu, Vetter et al. (1998) [37] use probabilities of events to define the profile, Lane and Brodley (1998) [38], Forrest et al (1996) [39] and Kosoresow and Hofineyr (1997) [40) use similarityof sequences that can be interpreted in a probabilistic framework

Unsupervised methods are use:fiıl ili applications where there is no prior knowledge as to the particular class of observations in a <lata set. For example, we may not be able to know for sure which transactions in a database are fraudulent and which are Iegitimate. In these situations, unsupervised methods can be used to :find groups or :find outliers in the data. Essentially, we collect data to provide a summary ofthe system that we are studying. Once we have a summary of the behaviour of the system, we can identify those observations that do not fit in with this behaviour, i.e. anomalous observations. This is our aim in using unsupervised statistical techniques for fraud detection.

The most popular unsupervised method used in <lata mining is clustering. This technique is used to :find natural groupings of observations in the data and is especially useful in market segmentation, However, cluster analysis can suffer from a bad choice of metric (the way we scale, transform cırıci combine variables to measure the 'distance' between observations); for example, it can be difficult to combine categorical and continuous variables in a good clustering metric. Observations may cluster differently on some subsets of variables than they do on others so that we may have more than one valid clustering in a <lata set.

We can use unsupervised methods such as clustering to help us form local models from which we can find local outliers in the <lata. In the context of fraud detection, a global outlier is a transaction anomalous to the entire data set; for example, a purchase of several thousand pounds would be a global outlier if all other transactions in the database were considerably less than that amount. Local outliers describe transactions that are anomalous when compared to subgroups of the data. Local outlier detection is effective in situations where the population is heterogeneous; this is true of credit card transaction data where spending behaviour between accounts can vary according to amounts spent and the purchases that are made. If we can identify the spending behaviour of a particular account, then a transaction is a local outlier if it is anomalous to spending in that account (or accounts similar to it), but not necessarily anomalous to the entire population of transactions. For example, a transaction ofa thousand pounds in an account where, historically, all transactions have been under a hundred pounds might be considered as a local outlier; however, such a transaction may not have been considered unusual if it had occurred in a high spending account, and thus would not be a global outlier,

The fundamental challenge is in the formation of the local model, which can be achieved in a variety of ways. One way is through cluster analysis. Here, legitimate transactions from all accounts are clustered into groups with similar characteristics. The local model, or profile, of a particular account is then determined by the clusters to which its transactions are allocated. If a future transaction from the account is then allocated to a cluster not in the account profile, then an alarm is raised for that transaction. Care must be exercised in choosing variables and metrics on which to cluster.

Nearest-neighbour methods can be employed to combine transaction information .from accounts that exhibit similar behaviour. We have developed Peer Group Analysis as a tool that uses local models of spending behaviour over time to detect changes in spending within accounts; we describe an application of Peer Group Analysis to fraud detection below. We follow this with a description of Break Point Analysis. Here, a local model is created and updated by drawing irn;brmationfrom transactions within the same account. Sequences of transactions within that account are compared with this local model to indicate changes in spending behaviour.

4.6 **Summary**

A fraud detection system attempts to discover illegitimate behavior. Fraud detection involves identifying fraud as quickly as possible once it has been perpetrated once fraud prevention has failed. There are varied applications of fraud detection techniques, in this chapter we have concentrated on credit card fraud detection using neural networks. As we have considered in the sections above the emphasis on fraud detection is supervised techniques and unsupervised credit card fraud detection has not received attention in the literature. And in this chapter we have demonstrated the unsupervised fraud detection and some of their application.

CONCLUSION

Neural networks are developed with the goal of modeling information processing and learning in the brain applied to a number of practical applications in various fields, including computational molecular biology.

Artificial neural networks are one of the promises for the future in computing. They offer an ability to perform tasks outside the scope of traditional processors. They can recognize patterns within vast <lata sets and then generalize those patterns into recommended courses ofaction. Neural networks learn, they are not programmed.

Yet, even though they are not traditionally programmed, the designing of neural networks does require a skill. It requires an "art." This art involves the understanding of the various network topologies, current hardware, current software tools, the application to be solved, and a strategy to acquire the necessary <lata to train the network. This art further involves the selection of learning rules, transfer functions, summation functions, and how to connect the neurons within the network.

Then, the art of neural networking requires a lot of hard work as <lata is fed into the system, performances are monitored, processes tweaked, connections added, rules modified, and on and on until the network achieves the desired results.

These desired results are statistical in nature. The network is not always right. It is for that reason that neural networks are finding themselves in applications where humans are also unable to always be right, Neural networks can now pick stocks, cull marketing prospects, approve loans, deny credit cards, tweak control systems, grade coins, and inspect work.

Yet, the future holds even more promises. Neural networks need faster hardware. They need to become part of hybrid systems which also utilize fuzzy logic and expert systems. It is then that these systems will be able to hear speech, read handwriting, and formulate actions. They will be able to become the intelligence behind robots that never tire nor become distracted. It is then that they will become the leading edge in an age of "intelügent" machines.

The purpose of this project was to represent an understanding to the broad subject of neural networks explaining the implementations of neural networks.

Chapter one described a general introduction of neural networks, the definition of artificial neural and the history of neural networks from 1940s when the first neuron

was developed. The differences between neural computing and traditional computing were presented. Also it was explained how neural networks are being used and where the future of neural networks technology may lie.

Chapter two was about neural networks architectures and algorithms. Single-layer and multilayer feedforward networks, recurrent networks and radial basis function networks were described. Supervised and unsupervised learning were also explained.

Chapter three was aimed to present real applications to let the reader to enter the world of neural networks as they are used. Neural networks applied in vast amounts of field, in medicine, business, pattern recognition, image compression arts and telecommunications. These applications were discussed.

Chapter four was aimed to show the important application of neural networks in fraud detection concentrating on credit card fraud detection and how to use unsupervised neural networks in fraud detection.

REFERENCES

[I) McCulloch, W. S. and Pitts, W. H. *"A logical calculus of the ideas immanent in nervous activity",* Bulletin of Mathematical Biophysics, 5: 115-133, .1943.

[2) Rosenblatt, F.Rosenblatt, *"The perceptron: A probabillstic model for information storage and organizaüon in the brain",* Psychological Review, 65:386-408, 1958.

[3] Anderson, J. A., and Rosenfeld, E. (Eds.), *"Neurocompuüng: Foundations of Research",* Cambridge, MA: MIT Press, 1988

[4] Selfridge, O. G., "Pandemonium: a paradigm for learning. Mechanisation of Thought Processes", *Proceedings ofa Symposium Held at the National Physical Laboratory,* 1958.

[5] Widrow, B and Hoff, "Adaptive Switching Circuits", *In 1960 !RE WESCON Convention Record,* pages 96 - 104. IRE., 1960.

[6] Minsky, M. and Papert, S., *"Perceptrons",* MIT Press, Cambridge. 1969

[7] Werbos, P.J., *"The Roots of Back propagaiion",* NY: John Wiley & Sons, 1974/1994.

[8] Parker, D.B., *"Leaming-Logic";* MIT Center for Computational Research in Economic and Management Science, Cambridge, MA, 1985

[9) Rumelhart, D., J. McClelland & the POP Research Group, *"Parallel Distributed Processing: Explorations in the Microstructure of Cognition V. 112",* MIT Press, Cambridge, MA, 1986.

[10) Churchland, P. and Sejnowski, T, *"The Computational Brain",* MIT Press Cambridge, 1992.

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

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

[13) Kohonen, T., "An adaptive associative memory principle". *IEEE Transactions on Computers,* C-23:444-445, 1974.

[14) Hebb, D. O., *"The Organization of Behavior",* Wiley, 1949.

[15) Hopfield, J. J., "Neural networks and physical systems with emergent computational abilities", *Proceedings of the National Academy of Sciences,* 79:2554, 1982

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

[17] Scheff, K. and. Szu, H. "Gram-Schmidt Orthogonalization Neural Networks for Optical Character Recognition", *Journal ofNeural Network Computing,* Winter, 1990.

[18] Paul Watta, Brijesh Desaie, Norman Dannug, Mohamad Hassoun, *"Image Compression using Backprop",* 1996.

[19] Schalkoff, R. J., *"Pattem Recognition: Statistical, Structural, and Neural Approaches",* John Wiley & Sons, New York, NY, 1992.

[20] Hutchison, W.R. & Stephens, K.R., "The Airline Marketing Tactician (AMT): a commercial application of adaptive networking". *Proceedings of the first lnternational Conference on Neural Networks,* 4: 753-756. IEEE Press. 1987.

[21] Robert Hecht-Nielsen., *"Neurocomputing",* Addison-Wesley, 1989

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

[23] K. Pope, "Technology Improves on the Nose As Science Tries to Imitate Smell", *Wal! Street Journal,* pp. Bl-2, 1995.

(24] P.E. Keller, R.T. Kouzes, L.J. Kangas, and S. Hashem, *"Transmission of Olfactory Information for Telemedicine",* IOS Press, Amsterdam,, pp. 168-172, 1995.

[25] Pacific Northwest National Laboratory (PNNL) "http://www. emsl. pnl. gov:2080/proj/neuron/neural ".

[26] Todd, P.M., and D.G. Loy (Eds.), *"Music and Connectionism".* Cambridge, MA: MIT Press, 1991

[27] Lippmann, R.P., "Review of neural networks for speech recognition". Neural Computation, l 1-38, 1989

[28] Brause, R., LangsdorfT. and Hepp M., "Neural Data Mining for Credit Card Fraud Detection", *Proceedings. 11th IEEE lnternational Conference on Tools with Artificial lntelligence,* 1999.

[29] Hassibi, K., *"Detecting Payment Card Fraud with Neural Networks. Business Applications of Neural Networks".* P.J.G. Lisboa, A.Vellido, B.Edisbury Eds. Singapore: World Scientific, 2000.

[30] Hand D.J. and Henley W.E .. "Statistical classification methods in consumer credit scoring: a review". *Journal of the Royal Statistical Society,* Series A, 160,523-41, 1997

[31] A. Classe "Caught in the neural net (Credit card fraud detection)" Accountancy, vol. 115, pp. 58-59, 1995.

[32] S. Ghosh and D. L. Reilly "Credit card fraud detection with a neural network", *in Proc. 27th Hawaii lnt. Conf Syst. Sci.,* pp. 621-630. 1994.

[33] "Canada Trust ink Falcon pact with HNC", Ai Expert, Seotemoer 1994.

[34] "Visa cracks down on fraud: Credit card ID systems t added muscle", Infbrmation Week, August 1996.

[35] J. Dorronsoro, F. Ginel, C. Sanchez, C. Santa Cruz, "Neural Fraud .ui;;Lci.A ion in Credit Card Operations", IEEE Trans. Neural Networks, pp. 827-834. 1997

[36] Aleskerov, E., Freisleben B., and Rao B., '°A Neural Network Based .uauiui:1:sc Mining System for Credit Card Fraud Detection", *Computational lntelligence for Financial Engineering, Proceedings ofthe IEEEIIAFE,* pp.220-226. 1997

[37] Qu, D., Vetter B. M., Wang F., Narayan R., Wu S. F., Hou Y. F., Gong F. and Sargor C. "Statistical Anomaly Detection for Link-State Routing Protocols". *Proceedings. Sixth Jnternational Conference on Network Protocols,* 62-70, 1998.

[38] Lane, T. and Brodley C. E., "Temporal Sequence Learning and Data education for Anomaly Detection". *Proceedings of the 5th ACM Conference on Computer and Communications Security (CCS-98).* New York, 150-158, 1998.

[39] Forrest, S., Hofmeyr S., Somayaji A. and LongstaffT. (1996). "A sense of selffor unix processes". *Proceedings of the 1996 IEEE Symposium on Security and Privacy, Los Alamitos, CA, 1996.*

[40] Kosoresow, A. P. and Hofineyr S. A., *"Intmsion Detection via System Cali Traces".* IEEE Software 14(5), 24-42, 1997.

[41] Lapedes, 87, Lapedes, A., and Farber, R., *"Non-Linear Signal Processir,ıg Using Neural Networks: Prediction and System Modeling",* Los Alamos National Lahoratory Report LA-UR-87-2662, 1987.