NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

NEURAL MEDICAL SYSTEMS

Graduation Project

COM – 400

Student: Haidar Alagha

Supervisor: Assoc. Prof. Dr. Adnan Khashman

Nicosia - 2002

ACKNOWLEDGMENT

First I would be honored to direct my thank to my supervisor Dr Adnan Khashman for being so co-operative and for his advices during me preparation to the graduation project.

As I would like to thank my family for giving me the chance to complete my academic study and specially I would like to thank my parents for supporting me and giving me the opportunity to achieve my goal in life.

Also I would like to thank all of the teachers with no exceptions for being so patients and for what they have taught us, specially Mr Tayseer Al-Shanablah, Dr Faieq Radwan, Dr Rahib and Miss Besime, and it is my privilege to thank the Associated Professor Dr Senol Bektas for standing beside us and helping us .

Finally I want to thank all my friends who helped and advised me during my preparation to the graduation project.

ABSTRACT

The world has become so different, it is now much more developed than any time before, looking everywhere we can notice development in many fields; seeking behind all of that, we will recognize that a lot of things have played great rule and we will notice that Artificial Neural Network will come on the top of all these technology.

The principle behind the A.N.N. is to simulate and make decision just as the brain does, applying this concept by using both of machines and softwares.

Well, the objectives of this project are to concentrate on the benefit of applying N.N. to various fields and also to focus on some problems that face or have faced.

As A.N.N. learns by examples so, A.N.N. can be trained to solve the most difficult problems in many applications and especially in medicine.

Artificial Neural Network has its own mark in many fields beside medicine, such as it goes in military and image analysis, business, industry and many other fields. Thus, neural network have been applied to medicine to help the physician of handling huge quantities of data to diagnose diseases, drug development and others.

TABLE OF CONTENTS

INTRODUCTION

Artificial Neural Network (A.N.N.) is one of the most effective weapon in the world of technology, so the A.N.N. can be found in both fields whether it is peaceful or military fields, the concept behind A.N.N. can be identify as an information processing paradigm, implemented in both of hardwares and softwares that is modeled after the biological processes of the brain. An A.N.N. is made up of a collection of highly interconnected nodes, called neurons or processing elements.

So here the A.N.N. was and still so powerful mean for reducing or even surmount on most of the problems that may exist in our world. Some of these fields where A.N.N. good at can be found in decision making, diagnostic analysis, image analysis, intelligent systems and some other more fields. Notice that N.Ns are so powerful that it has been replaced the human in all of these fields.

Chapter one describes the history of the Artificial Neural Network and how does it simulate the brain, also this chapter described the advantages, disadvantages and some of the problems that A.N.Ns may have.

Chapter two which describes the architecture of the Artificial Neural Network, and the ways that N.N. can be trained with, which are the supervised and unsupervised learning methods, and some other applications that A.N.N. involved with. As I also describe the next development in N.N.

Chapter three: this chapter will be specialized in the medical applications and describes some fields where we can find the N.N. in medicine, whether in medical diagnostic aides, biochemical analysis or in medical image analysis.

Chapter four discusses the segmentation of the medical images and how useful the neural network is in detecting diseases.

The aims of this project is to retrieve some information about the Artificial Neural Networks and its uses, history, and some applications where the A.N.Ns goes. As also to focus on some applications that Artificial Neural Networks involve in, such as: industrial applications, military applications, business issues, diagnostic analysis and medicine applications as well.

CHAPTER ONE

INTRODUCTION TO NEURAL NETWORKS

1.1 Overview.

Neural Networks are an information processing technique based on the way biological nervous systems, such as the brain, process information. The fundamental concept of neural networks is the structure of the information processing system. Composed of a large number of highly interconnected processing elements or neurons, a neural network system uses the human-like technique of learning by example to resolve problems. The neural network is configured for a specific application, such as data classification or pattern recognition, through a learning process called training. Just as in biological systems, learning involves adjustments to the synaptic connections that exist between the neurons.

Neural networks can differ on the way their neurons are connected; the specific kinds of computations their neurons do; the way they transmit patterns of activity throughout the network; and the way they learn including their learning rate. Neural networks are being applied to an increasing large number of real world problems. Their primary advantage is that they can solve problems that are too complex for conventional technologies -- problems that do not have an algorithmic solution or for which an algorithmic solution is too complex to be defined. In general, neural networks are well suited to problems that people are good at solving, but for which computers generally are not. These problems include pattern recognition and forecasting -- which requires the recognition of trends in data.

1.2 History of Neural Networks.

The study of the human brain is thousands of years old. With the advent of modern electronics, it was only natural to try to harness this thinking process. The first step toward artificial neural networks came in 1943 when Warren McCulloch [1], a neurophysiologist, and a young mathematician, Walter Pitts, wrote a paper on how neurons might work. They modeled a simple neural network with electrical circuits.

As computers advanced into their infancy of the 1950s, it became possible to begin to model the rudiments of these theories concerning human thought. Nathanial Rochester from the IBM research laboratories led the first effort to simulate a neural network. That first attempt failed. But later attempts were successful. It was during this time that traditional computing began to flower and, as it did, the emphasis in computing left the neural research in the background.

Yet, throughout this time, advocates of "thinking machines" continued to argue their cases. In 1956 the Dartmouth Summer Research Project on Artificial Intelligence provided a boost to both artificial intelligence and neural networks. One of the outcomes of this process was to stimulate research in both the intelligent side, A.I., as it is known throughout the industry, and in the much lower level neural processing part of the brain.

In the years following the Dartmouth Project, John von Neumann suggested imitating simple neuron functions by using telegraph relays or vacuum tubes. Also, Frank Rosenblatt, a neuro-biologist of Cornell, began work on the Perceptron. He was intrigued with the operation of the eye of a fly. Much of the processing which tells a fly to flee is done in its eye. The Perceptron, which resulted from this research, was built in hardware and is the oldest neural network still in use today. A single-layer perceptron was found to be useful in classifying a continuous-valued set of inputs into one of two classes. The perceptron computes a weighted sum of the inputs, subtracts a threshold, and passes one of two possible values out as the result. Unfortunately, the perceptron is limited and was proven as such during the "disillusioned years" in Marvin Minsky and Seymour Papert's 1969 book Perceptrons. In 1959, Bernard Widrow and Marcian Hoff [2] of Stanford developed models they called ADALINE and MADALINE. These models were named for their use of Multiple Adaptive Linear Elements. MADALINE was the first neural network to be applied to a real world problem. It is an adaptive filter which eliminates echoes on phone lines. This neural network is still in commercial use. Unfortunately, these earlier successes caused people to exaggerate the potential of neural networks, particularly in light of the limitation in the electronics then available. This excessive hype, which flowed out of the academic and technical worlds, infected the general literature of the time. Disappointment set in as promises were unfilled. Also, a fear set in as writers began to ponder what effect "thinking machines" would have on man. Asimov's series on robots revealed the effects on man's morals and values when machines where capable of doing all of mankind's work. Other writers created more sinister computers, such as HAL from the movie 2001. These fears, combined with unfulfilled, outrageous claims, caused respected voices to critique the neural network research. The result was to halt much of the funding. This period of stunted growth lasted through 1981.

Today, neural networks discussions are occurring everywhere. Their promise seems very bright as nature itself is the proof that this kind of thing works. Yet, its future, indeed the very key to the whole technology, lies in hardware development. Currently most neural network development is simply proving that the principal works. This research is developing neural networks that, due to processing limitations, take weeks to learn. To take these prototypes out of the lab and put them into use requires specialized chips. Companies are working on three types of neuro chips - digital, analog, and optical. Some companies are working on creating a "silicon compiler" to generate a neural network Application Specific Integrated Circuit (ASIC). These ASICs and neuron-like digital chips appear to be the wave of the near future. Ultimately, optical chips look very promising. Yet, it may be years before optical chips see the light of day in commercial applications.

1.3 What are Artificial Neural Networks?

Artificial Neural Networks are relatively crude electronic models based on the neural structure of the brain. The brain basically learns from experience. It is natural proof that some problems that are beyond the scope of current computers are indeed solvable by small energy efficient packages. 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 math. But computers have trouble recognizing even simple patterns much less generalizing those patterns of the past into actions of the 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 does not utilize traditional programming but involves the creation of massively parallel networks and the training of those networks to solve specific problems. This field also utilizes words very different from traditional computing, words like behave, react, self-organize, learn, generalize, and forget.

Haykin, S. (1994), Neural Networks: A Comprehensive Foundation, NY: Macmillan [3].

1.4 Analogy to the Brain.

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

The power of the human mind comes from the sheer numbers of these basic components and the multiple connections between them. It also comes from genetic programming and learning.The individual neurons are complicated. They have a myriad of parts, sub-systems, and control mechanisms. They convey information via a host of electrochemical pathways. There are over one hundred different classes of neurons, depending on the classification method used. Together these neurons and their connections form a process which is not binary, not stable, and not synchronous. In short, it is nothing like the currently available electronic computers, or even artificial neural networks.

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

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 1.1 shows the relationship of these four parts.

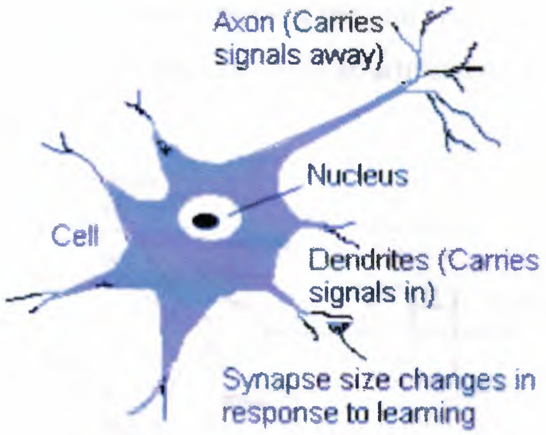

Axon (Carries signals away)
Nucleus
Cell
Dendrites (Carries signals in)
Synapse size changes in response to learning

Figure 1.1 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, nucleus, axon, and synapses. Dendrites are hair-like extensions of the nucleus which act like input channels. These input channels receive their input through the synapses of other neurons. The nucleus then processes these incoming signals over time. The nucleus then turns that processed value into an output which is sent out to other neurons through the axon and the synapses.

1.5 Artificial Neurons and How They Work

Recent experimental data 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 better understanding of neurons, and as technology advances, network designers can continue to improve their systems by building upon man's understanding of the biological brain. But currently, the goal of artificial neural networks is not the grandiose recreation of the brain. On the contrary, neural network researchers are seeking an understanding of nature's capabilities for which people can engineer solutions to problems that have not been solved by traditional computing. To do this, the basic unit of neural networks, the artificial neurons, simulate the four basic functions of natural neurons. Figure 1.2 shows a fundamental representation of an artificial neuron.

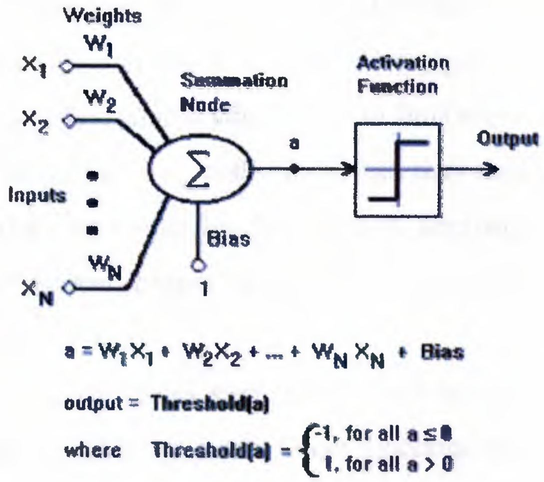


Figure 1.2 A Basic Artificial Neuron.

In Figure 1.2, various inputs to the network are represented by the mathematical symbol, x(n). Each of these inputs are multiplied by a connection weight. These weights are represented by w(n). In the simplest case, these products are simply summed, fed through a transfer function to generate a result, and then output. This process lends itself to physical implementation on a large scale in a small package. This electronic implementation is still possible with other network structures which utilize different summing functions as well as different transfer functions.

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

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

Other applications might simply sum and compare to a threshold, thereby producing one of two possible outputs, a zero or a one. Other functions scale the outputs to match the application, such as the values minus one and one. Some functions even integrate the input data over time, creating time-dependent networks. Valiant, L. (1988), "Functionality in Neural Nets," Learning and Knowledge [4].

1.6 The Ups and Downs of Neural Networks.

There are many good points to neural-networks and advances in this field will increase their popularity. There are excellent as pattern classifiers/recognizers - and can be used where traditional techniques do not work. Neural-networks can handle exceptions and abnormal input data, very important for systems that handle a wide range of data (radar and sonar systems, for example). Many neural networks are biologically plausible, which means they may provide clues as to how the brain works as they progress. Advances in neuroscience will also help advance neural networks to the point where they will be able to classify objects with the accuracy of a human at the speed of a computer! The future is bright, the present however...

Yes, there are quite a few down points to neural networks. Most of them, though, lie with our lack of hardware. The power of neural-networks lie in their ability to process information in a parallel fashion (that is, process multiple chunks of data simultaneously). Unfortunately, machines today are serial - they only execute one instruction at a time. Therefore, modeling parallel processing on serial machines can be a very time-consuming process. As with everything in this day and age, time is of the essence, which often leaves neural networks out of the list of viable solutions to a problem.

Other problems with neural networks are the lack of defining rules to help construct a network given a problem - there are many factors to take into consideration: the learning algorithm, architecture, number of neurons per layer, number of layers, data representation and much more. Again, with time being so important, companies cannot afford to invest to time to develop a network to solve the problem efficiently. This will all change as neural networking advances.

1.7 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 of neural networks.

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

Table 1.1 Comparison of Computing 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 Method	By rules (didactically)	ByExample (Socratically)
Applications	Accounting Word processing Math 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 occurs by the expert system generating the computer's programming itself, it does that through "programming" of its own. This programming is needed to establish the rules for a particular application. This method of establishing rules is also complex and does require a detail oriented person.

1.8 Advantage and Disadvantage of Neural Network:

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

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

1.9 Summary

A brief background about the A.N.N. since aneuronphysiologist and a mathematician that was in the 1943 and may be before and passing through the years 1950's , 1960's… till present days and A.N.N is in an extremely enormous development.

A Neural Network depends on its action to the imitation of the brain so, the biologist trying to simulate the action and the reaction of the brain, just as the human learns from example here also the N.N learns by example.

A neural network can be used in many fields specifically in these where traditional techniques can not be applied, it can also handle the abnormal tasks where it has to be fed by too much data such as the radar system and the sonar system in the submarine under the water.

With all these developments in neural networks but it is also can not match the human kind's brain in pattern recognition or even an animal brain when it faces a similar problem.

CHAPTER TWO

THE STRUCTURE OF NEURAL NETWORKS

2.1 Architecture of Neural Networks.

In this section which is the Architecture of Neural Network, I am going to give a hint about how N.N. has been designed, and how the signals travel and what kind of networks does it have. Here Neural Network is been classified in two categories, which are Feed-forward Network and Feed-back Network.

2.1.1 Feed-forward Networks.

Feed-forward A.N.Ns (figure 2.1) allow signals to travel one way only; from input to output. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. Feed-forward A.N.Ns tend to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition. This type of organization is also referred to as bottom-up or top-down.

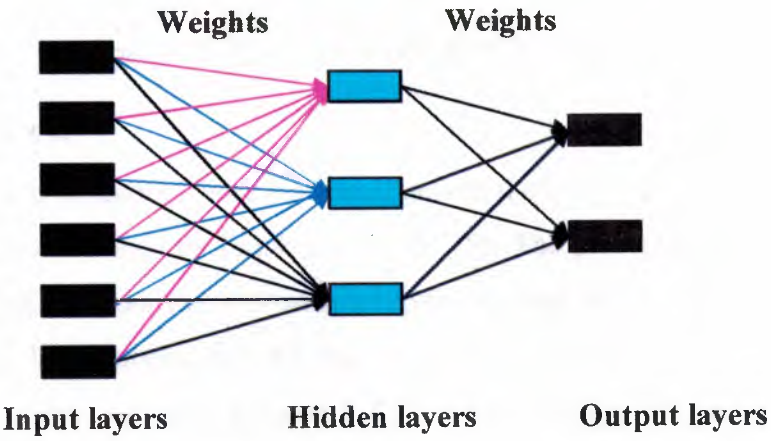


Figure 2.1 An example of a simple feedforward network

2.1.2 Feedback Networks.

Feedback networks (figure 2.2) can have signals travelling in both directions by introducing loops in the network. Feedback networks are very powerful and can get extremely complicated. Feedback networks are dynamic; their 'state' is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Feedback architectures are also referred to as interactive or recurrent, although the latter term is often used to denote feedback connections in single-layer organisations.

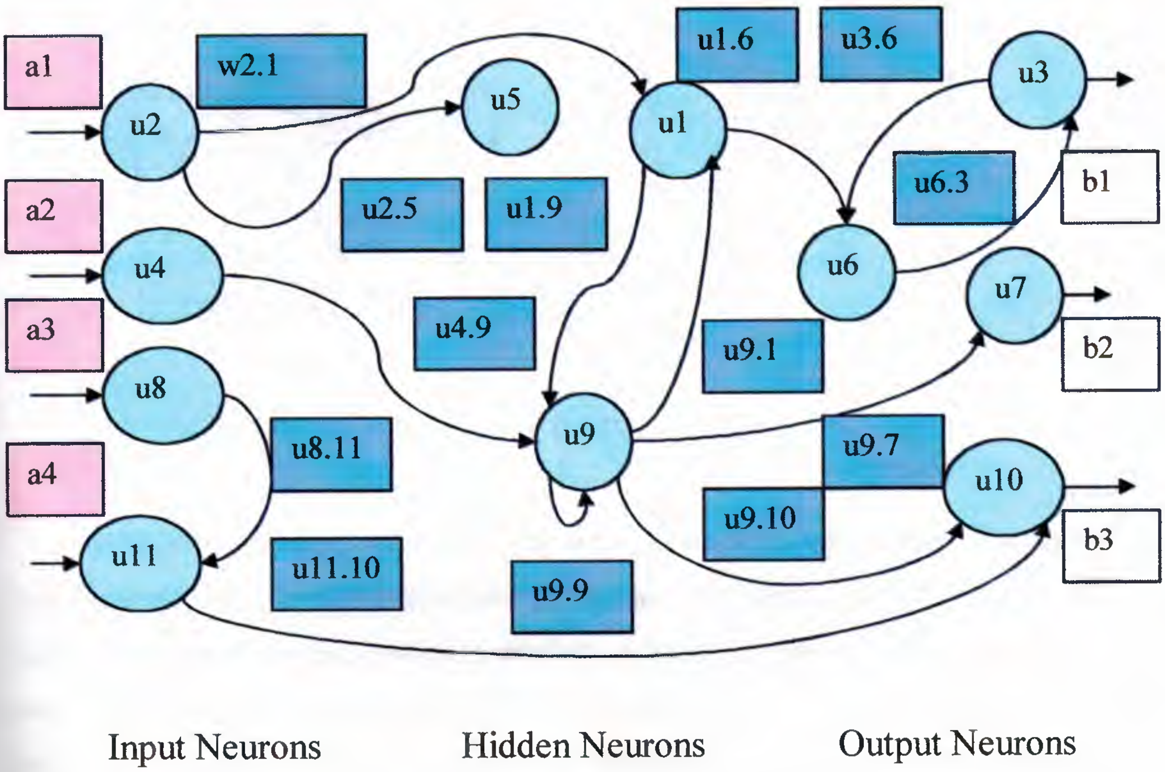


Figure 2.2 An example of a feedback network

2.2 Perceptrons.

The most influential work on neural nets in the 60's went under the heading of 'perceptrons' a term coined by Frank Rosenblatt [5]. The perceptron (figure 2.3) turns out to be an MCP model (neuron with weighted inputs) with some additional, fixed, pre--processing. Units labeled A1, A2, Aj, Ap are called association units and their task is to extract specific, localised featured from the input images. Perceptrons mimic the basic idea behind the mammalian visual system. They were mainly used in pattern recognition even though their capabilities extended a lot more.

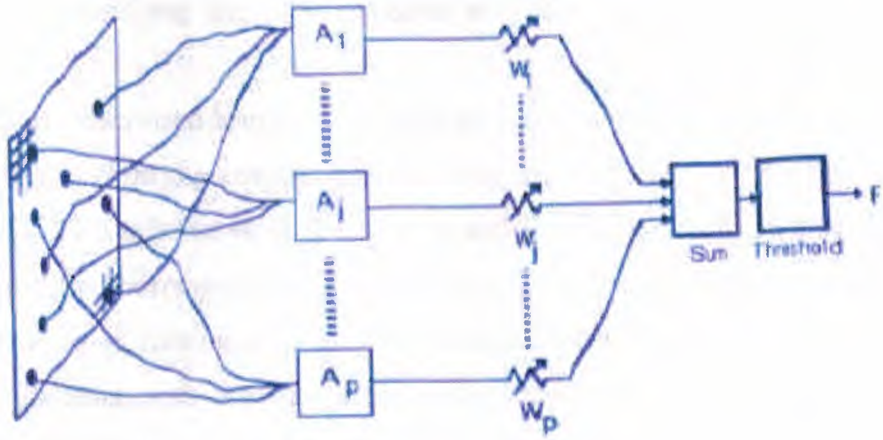


Figure 2.3 The Perceptron

In 1969 Minsky and Papert [6] wrote a book in which they described the limitations of single layer Perceptrons. The impact that the book had was tremendous and caused a lot of neural network researchers to loose their interest. The book was very well written and showed mathematically that single layer perceptrons could not do some basic pattern recognition operations like determining the parity of a shape or determining whether a shape is connected or not. What they did not realised, until the 80's, is that given the appropriate training, multilevel perceptrons can do these operations.

2.3 Teaching an Artificial Neural Network.

In this part of teaching the Neural Networks we will go through the learning methods of N.N., Neural Networks can be classified according to the way they learn into two kinds of learning methods which are SUPERVISED LEARNING and UNSUPERVISED LEARNING.

2.3.1 Supervised Learning.

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 next iteration, or cycle, 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 data to the network. This data 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 data must be used. This training phase can consume a lot of time. In prototype systems, with inadequate processing power, learning 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 learn 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 data. 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 result, the system has to learn everything together, finding the best weight settings for the total set of facts. For example, in teaching a system to recognize pixel patterns for the ten digits, if there were twenty examples of each digit, all the examples of the digit seven should not be presented at the same time.

How the input and output data is represented, or encoded, is a major component to successfully instructing a network. Artificial networks only deal with numeric input data. Therefore, the raw data must often be converted from the external environment. Additionally, it is usually necessary to scale the data, 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 data 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 data but has learned the general patterns involved within an application.

2.3.2 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 is 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 back-end 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. These networks look for regularities or trends in the input signals, and makes adaptations according to the function of the network. Even without being told whether it's right or wrong, 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 learning 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 amplify 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 government because military.

Situations often do not have a data set available to train a network until a conflict arises.

2.3.3 Learning Rate.

The rate at which A.N.N's 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 of both worlds.

Generally, several factors besides time have to be considered when discussing the off-line training task, which is often described as "tiresome." Network complexity, size, paradigm selection, architecture, type of learning rule or 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 learning functions have some provision for a learning rate, or learning constant. Usually this term is positive and between zero and one. If the learning rate is greater than one, it is easy for the learning algorithm to overshoot in correcting the weights, and the network will oscillate. Small values of the learning rate will not correct the current error as quickly, but if small steps are taken in correcting errors, there is a good chance of arriving at the best minimum convergence.

2.3.4 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 learning as their main objective. Others are experimenting with adaptations of their perceptions of how nature handles learning. Either way, man's understanding of how neural processing actually works is very limited. Learning is certainly more complex than the simplifications represented by the learning laws currently developed. A few of the major laws are presented as examples.

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

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

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

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

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

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

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

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

Further, the size of the 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-organizing maps or self-organizing topologies. Valiant, L. (1988), "Functionality in Neural Nets," Learning and Knowledge [7].

2.4 Artificial Network Operations.

The other part of the "art" of using neural networks revolve around the myriad of ways these individual neurons can be clustered together. This clustering occurs in the human mind in such a way that information can be processed in a dynamic, interactive, and self-organizing way. Biologically, neural networks are constructed in a three-dimensional world from microscopic components. These neurons seem capable of nearly unrestricted interconnections. That is not true of any proposed, or existing, man-made network. Integrated circuits, using current technology, are two-dimensional devices with a limited number of layers for interconnection. This physical reality restrains the types, and scope, of artificial neural networks that can be implemented in silicon.

Currently, neural networks are the simple clustering of the primitive artificial neurons. This clustering occurs by creating layers which are then connected to one another. How these layers connect is the other part of the "art" of engineering networks to resolve real world problems.

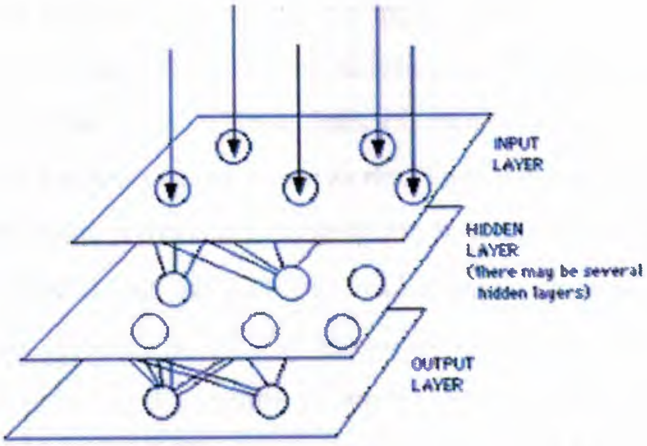


Figure 2.4 A Simple Neural Network Diagram.

Basically, all artificial neural networks have a similar structure or topology as shown in Figure 2.4 In that structure some of the neurons interfaces to the real world to receive its inputs. Other neurons provide the real world with the network's outputs. This output might be the particular character that the network thinks that it has scanned or the particular image it thinks is being viewed. All the rest of the neurons are hidden from view. But a neural network is more than a bunch of neurons. Some early researchers tried to simply connect neurons in a random manner, without much success. Now, it is known that even the brains of snails are structured devices. One of the easiest ways to design a structure is to create layers of elements. It is the grouping of these neurons into layers, the connections between these layers, and the summation and transfer functions that comprises a functioning neural network. The general terms used to describe these characteristics are common to all networks.

Although there are useful networks which contain only one layer, or even one element, most applications require networks that contain at least the three normal types of layers - input, hidden, and output. The layers of input neurons receive the data either from input files or directly from electronic sensors in real-time applications. The output layer sends information directly to the outside world, to a secondary computer process, or to other devices such as a mechanical control system. Between these two layers can be many hidden layers. These internal layers contain many of the neurons in various interconnected structures. The inputs and outputs of each of these hidden neurons simply go to other neurons.

In most networks each neuron in a hidden layer receives the signals from all of the neurons in a layer above it, typically an input layer. After a neuron performs its function it passes its output to all of the neurons in the layer below it, providing a feedforward path to the output. These lines of communication from one neuron to another are important aspects of neural networks. They are the glue to the system. They are the connections which provide a variable strength to an input. There are two types of these connections. One causes the summing mechanism of the next neuron to add while the other causes it to subtract. In more human terms one excites while the other inhibits.

Some networks want a neuron to inhibit the other neurons in the same layer. This is called lateral inhibition. The most common use of this is in the output layer. For example in text recognition if the probability of a character being a "P" is .85 and the probability of the character being an "F" is .65, the network wants to choose the highest probability and inhibit all the others. It can do that with lateral inhibition. This concept is also called competition.

Another type of connection is feedback. This is where the output of one layer routes back to a previous layer. An example of this is shown in Figure 2.5

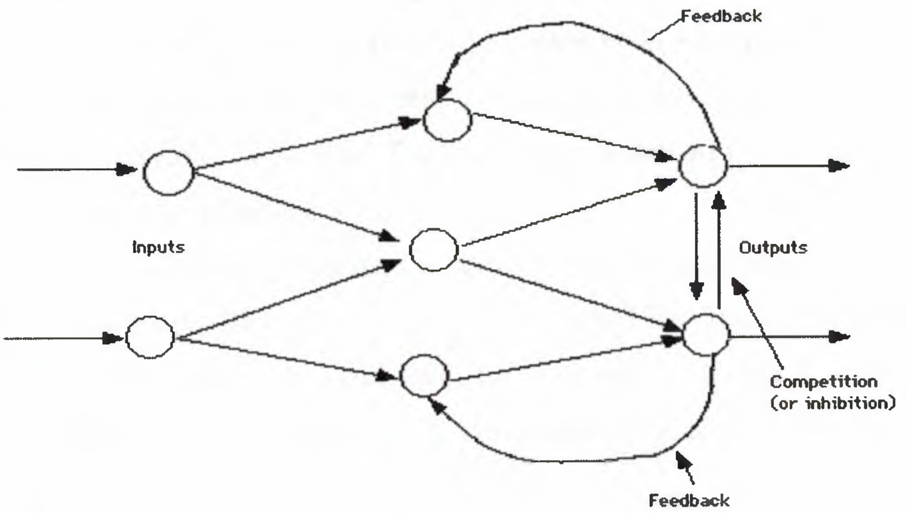


Figure 2.5 Simple Network with Feedback and Competition.

The way that the neurons are connected to each other has a significant impaction the operation of the network. In the larger, more professional software development packages the user is allowed to add, delete, and control these connections at will. By "tweaking" parameters these connections can be made to either excite or inhibit

2.5 A Description of The Back Propagation Algorithm.

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

The back-propagation algorithm is easiest to understand if all the units in the network are linear. The algorithm computes each EW by first computing the EA, the rate at which the error changes as the activity level of a unit is changed. For output units, the EA is simply the difference between the actual and the desired output. To compute the EA for a hidden unit in the layer just before the output layer, we first identify all the weights between that hidden unit and the output units to which it is connected. We then multiply those weights by the EAs of those output units and add the products. This sum equals the EA for the chosen hidden unit. After calculating all the EAs in the hidden layer just before the output layer, we can compute in like fashion the EAs for other layers, moving from layer to layer in a direction opposite to the way activities propagate through the network. This is what gives back propagation its name. Once the EA has been computed for a unit, it is straight forward to compute the EW for each incoming connection of the unit. The EW is the product of the EA and the activity through the incoming connection.

Note that for non-linear units, the back-propagation algorithm includes an extra step. Before back-propagating, the EA must be converted into the EI, the rate at which the error changes as the total input received by a unit is changed. According to Haykin, S. (1994), Neural Networks: A Comprehensive Foundation, NY: Macmillan, p. 2: [8].

2.5.1 A Back-Propagation Network Example.

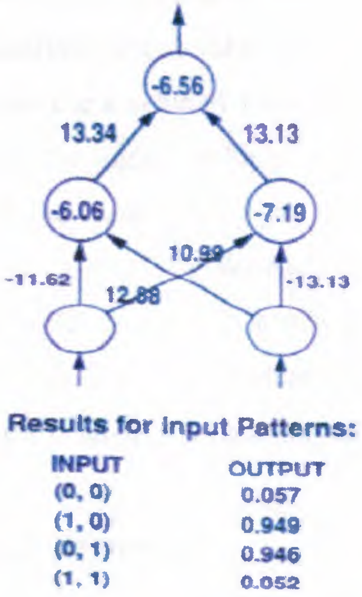
In this example a back-propagation network would be used to solve a specific problem, that one of an X-OR logic gate. That means that patterns of (0,0) or (1,1) should produce a value close to zero in the output node, and input patterns of (1,0) or (0,1) should produce a value near one in the output node as in Figure 2.5.

Finding a set of connection weights for this task is not easy; it requires application of the back-propagation algorithm for several thousand iterations to achieve a good set of connection weights and neuron thresholds.

The basic architecture for this problem has two input nodes, two hidden nodes, and a single output node as shown in Figure 2.6. This structure has variable thresholds on the two hidden and one output node (unit). This means that there are a total of 9 variables in the system:

- Four weights connecting the input to the hidden nodes
- Two weights connecting the hidden to the output node
- Three thresholds

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



Results for Input Patterns:

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

Figure 2.6 Back-propagation network

Now we move our attention to the next layer up. For each neuron in this layer, we calculate an input which is the weighted sum of all the activations from the first layer. The weighted sum is achieved by vector multiplying the activations in the first layer by a "connection matrix". In our case we get a value of 0*(-11,62)+ 1*(10,99) = 10,99 for the neuron on the left in the second layer, and 0*(12,88) +1*(13,13) =13,13 for the neuron on the right.

These are not the activation of these neurones, though. To obtain the activations, we add a "threshold" value (which is found for each neuron using the back-propagation rule), and apply an input-output (transfer) function. The transfer function is defined for each different network. In our case it is a sigmoid (figure 2.7):

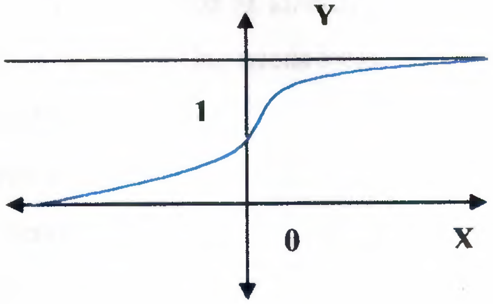


Figure 2.7 a transfer function

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

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

2.6 Applications of Neural Networks.

As most of us know that Neural Network has been involved in too many fields, I mean that N.N. is being used in a lot of branches. What I am going to cover here in this section are some applications of N.N., but I am going to focus on three applications here which are Neural Networks in Practice which covers (Sales forecasting, Industrial process control …etc), in Medicine and in Business.

2.6.1 Neural Networks in Practice.

Given this description of neural networks and how they work, what real world applications are they suited for? Neural networks have broad applicability to real world business problems. In fact, they have already been successfully applied in many industries. Since neural networks are best at identifying patterns or trends in data, they are well suited for prediction or forecasting needs including:

- Sales forecasting
- Industrial process control
- Customer research
- Data validation
- Risk management
- Target marketing

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

2.6.2 Neural Networks in Medicine.

Artificial Neural Networks (A.N.N.) is 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 modeling parts of the human body and recognizing diseases from various scans (e.g. cardiograms, CAT scans, ultrasonic scans, etc.).

Neural networks are ideal in recognizing diseases using scans 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 recognize 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 if the system is to perform reliably and efficiently.

I am going to discuss that part in details in the next two chapters.

2.6.3 Neural Networks in Business.

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

2.6.3.1 Marketing.

There is a marketing application which has been integrated with a neural network system. The Airline Marketing Tactician (a trademark abbreviated as AMT) is a computer system made of various intelligent technologies including expert systems. A feedforward neural network is integrated with the AMT and was trained using back-propagation to assist the marketing control of airline seat allocations. The adaptive neural approach was amenable to rule expression. Additionally, the application's environment changed rapidly and constantly, which required a continuously adaptive solution. The system is used to monitor and recommend booking advice for each departure. Such information has a direct impact on the profitability of an airline and can provide a technological advantage for users of the system.
Hutchison & Stephens, 1987 [9].

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

2.7 What can you do with an N.N. and what not?

In principle, N.Ns can compute any computable function, i.e., they can do everything a normal digital computer can do, (Valiant, 1988; Siegelmann and Sontag, 1999; Orponen, 2000; Sima and Orponen, 2001) [10], or perhaps even more, under some assumptions of doubtful practicality.

Practical applications of N.Ns most often employ supervised learning. For supervised learning, you must provide training data that includes both the input and the desired result (the target value). After successful training, you can present input data alone to the N.N. (that is, input data without the desired result), and the N.N. will compute an output value that approximates the desired result. However, for training to be successful, you may need lots of training data and lots of computer time to do the training. In many applications, such as image and text processing, you will have to do a lot of work to select appropriate input data and to code the data as numeric values.

In practice, N.Ns are especially useful for classification and function approximation/mapping problems which are tolerant of some imprecision, which have lots of training data available, but to which hard and fast rules (such as those that might be used in an expert system) cannot easily be applied. Almost any finite-dimensional vector function on a compact set can be approximated to arbitrary precision by feedforward N.Ns (which are the type most often used in practical applications) if you have enough data and enough computing resources.

To be somewhat more precise, feedforward networks with a single hidden layer and trained by least-squares are statistically consistent estimators of arbitrary square-integrable regression functions under certain practically-satisfiable assumptions regarding sampling, target noise, number of hidden units, size of weights, and form of hidden-unit activation function. Such networks can also be trained as statistically consistent estimators of derivatives of regression functions and quintiles of the conditional noise distribution. Feedforward networks with a single hidden layer using threshold or sigmoid activation functions are universally consistent estimators of binary classifications under similar assumptions. Note that these results are stronger than the universal approximation theorems that merely show the existence of weights for arbitrarily accurate approximations, without demonstrating that such weights can be obtained by learning.

Unfortunately, the above consistency results depend on one impractical assumption: that the networks are trained by an error (or misclassification rate) minimization technique that comes arbitrarily close to the global minimum. Such minimization is computationally intractable except in small or simple problems. In practice, however, you can usually get good results without doing a full-blown global optimization; e.g., using multiple (say, 10 to 1000) random weight initializations is usually sufficient.

One example of a function that a typical neural net cannot learn is Y=1/X on the open interval (0,1). An open interval is not a compact set. With any bounded output activation function, the error will get arbitrarily large as the input approaches zero. Of course, you could make the output activation function a reciprocal function and easily get a perfect fit, but neural networks are most often used in situations where you do not have enough prior knowledge to set the activation function in such a clever way. There are also many other important problems that are so difficult that a neural network will be unable to learn them without memorizing the entire training set, such as:

- Predicting random or pseudo-random numbers.
- Factoring large integers.
- Determing whether a lahapklsrge integer is prime or composite.
- Decrypting anything encrypted by a good algorithm.

And it is important to understand that there are no methods for training NNs that can magically create information that is not contained in the training data.

2.8 What the Next Developments Will Be?

The vendors within the industry predict that migration from tools to applications will continue. In particular, the trend is to move toward hybrid systems. These systems will encompass other types of processes, such as fuzzy logic, expert systems, and kinetic algorithms. Indeed, several manufactures are working on "fuzzy neurons." The greatest interest is on merging fuzzy logic with neural networks. Fuzzy logic incorporates the inexactness of life into mathematics. In life most pieces of data do not exactly fit into certain categories. For instance, a person is not just short or tall. He can be kinda short, pretty tall, a little above average, or very tall. Fuzzy logic takes these real-world variations into account. In potential application of neural networks, in systems which solve real problems, this fuzziness is a large part of the problem. In automating a car, to stop is not to slam on the brakes, to speed up is not to "burn rubber." To help neural networks accomodate this fuzziness of life, some researchers are developing fuzzy neurons. These neurons do not simply give yes/no answers. They provide a more fuzzy answer.

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

2.9 Summary

In this chapter I tried to concentrate on the architecture and the structure of the neural network, here the Neural Networks are able to act in two ways which are feedback and feedforward neural network.

Feedforward N.N means it can only travel in one way (with no loop return to itself).

Feedback N.N has a loop which means it can travel in both directions.

And after that I have moved ahead to give a hint about how the N.N. can be trained. Well, according to my study and research, I have noticed that the neural net. is able to be trained by two methods.

The First one is Supervised Learning which means that N.N. has to be given both of the input and the output, and also the data which according to it will act and make decisions.

The second one is the Unsupervised learning, here the Neural Network durring the learning process requires only the input data vector to train it.

So, we can find N.N. in so many branches such as in Business, Industry, Forecast, Military, and Medicine.

CHAPTER THREE

NEURAL NETWORKS IN MEDICINE

3.1 Medical Informatics

Diagnosis Prediction via an Artificial Neural Network Knowledge Base.

The use of expert systems as means of predicting medical diagnosis and recommending successful treatments has been a highly active research field in the past few years. Development of medical expert systems that use artificial neural networks as their knowledge bases appears to be a promising method for predicting diagnosis and possible treatment routine. The network may in the future serve as a knowledge base for an expert system specializing in medical diagnosis, testing evaluation, treatment evaluation, and treatment effectiveness. The project serves as the first component of a much larger system that will assist physicians facilitate the reasonable ordering tests and treatments and minimize unnecessary laboratory routines while reducing operational costs.

The network correctly classified 965 out of 1292 cases (74.7%) in the training set and 418 out of 554 cases (75.5%) in the testing set. In some cases of classification, the network's prediction appears to be reasonable even though they differ from the physician's diagnoses. The notion of "reasonable" can be implied if the treatment, need for further diagnostic testing, clinical follow-up, and outcome are equivalent. For instance, muscle strain was sometimes incorrectly diagnosed as abdominal pain if abdomen was the primary body part. If we accept these "reasonable" diagnoses, network performance increases by 6.5%.

These preliminary results demonstrate the viability of a neural network as a knowledge base. Network performance and robustness would be increased if additional data were available.

One of the major applications of medical informatics has been the implementation and use of expert systems to predict medical diagnoses based upon a set of symptoms. Furthermore, such expert systems serve as an aid to medical professionals in recommending effective laboratory testing routines and effective treatments. An intelligent computer program assisting medical diagnosis offers the physician easy access to a wealth of information from past patient data. Such a resource may help hospitals reduce excessive costs from unnecessary laboratory testing and ineffective patient treatment, while maintaining high quality of medical care. So far, these expert systems only serve as aids to the physician and are not 100% reliable. The end goal is to develop an improved expert system. In our current investigation, we attempt to build the underlying infrastructure, or knowledge base, for such a system. Neural Network Learning and Expert Systems." Stephen I. Gallant, 1993 [11].

3.2 When Can Artificial Neural Networks Be Applied to Medicine?

The A.N.N. approach to the analysis of data will see extensive application to biomedical problems in the next few years. It has already been successfully applied to various areas of medicine, such as diagnostic aides, biochemical analysis, image analysis, and drug development.

3.2.1 Medical Diagnostic Aides.

The application of A.N.Ns in diagnosing heart attacks received publicity in the Wall Street Journal when the A.N.N. was able to diagnose with better accuracy than physicians. This application is significant because it was used in the emergency room where the physicians are not able to handle large amounts of data.

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

In the United Kingdom, an A.N.N. used in the early diagnosis of myocardial infarction is currently undergoing clinical testing at four hospitals. At the research level, A.N.Ns 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).

3.2.2 Biochemical Analysis.

A.N.Ns are used in a wide variety of analytical chemistry applications. In medicine, A.N.Ns have been used to analyze blood and urine samples, track glucose levels in diabetics, determine ion levels in body fluids, and detect pathological conditions such as tuberculosis. At Pacific Northwest National Laboratory, A.N.Ns are being combined with chemical sensor arrays and spectrometers for use in automated chemical analysis.

3.2.3 Medical Image Analysis.

A.N.Ns 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 micro-calcifications in mammograms, classification of chest x-rays, tissue and vessel classification in Magnetic Resonance Images (MRI), x-ray spectral reconstruction, determination of skeletal age from x-ray images, and determination of brain maturation. At Pacific Northwest National Laboratory, A.N.Ns are being developed to examine thallium centigram images of the heart and identify the existence of infarctions. Another project at Pacific Northwest National Laboratory uses A.N.N. technology to aid in the visualization of three-dimensional ultrasonic images.

3.2.4 Drug Development.

Researchers at the National Institutes of Health as well as other institutions have used A.N.Ns as tools in the development of drugs for treating cancer and AIDS. A.N.Ns are also used in the process of modeling biomolecules. At Pacific Northwest National Laboratory, A.N.N. technology is being incorporated into the software used to model protein molecules.

3.3 Investigation of the use of Neural Networks for Computerised Medical Image Analysis.

Advances in clinical medical imaging have brought about the routine production of vast numbers of medical images that need to be analysed. As a result an enormous amount of computer vision research effort has been targeted at achieving automated medical image analysis. This has proved to be an elusive goal in many cases. The complexity of the problems encountered has prompted considerable interest in the use of neural networks for such applications. However, many reports of such work have been unsatisfactory in that often only qualitative results are reported, or only few patient cases are used. This thesis presents a study of the use of neural networks and computer vision for medical image analysis which aims to quantitatively investigate and demonstrate the potential of neural networks in such an application. A medical image analysis problem was selected which would facilitate this. The problem chosen was the automatic detection of acoustic neuromas in MR images of the head.

Since neural networks excel at statistical pattern recognition tasks a broadly bottom-up approach to the problem was adopted. Neural networks were utilised for `intelligent' tasks which were supported by more conventional image processing operations in order to achieve the objectives set. The prototype system developed as a result of the study achieved a100% sensitivity and a 99.0% selectivity on a dataset of 50 patient cases. Shane Dickson. PhD thesis, Department of Computer Science, University of Bristol, February 1998 [12].

3.4 Modeling and Diagnosing the Cardiovascular System.

Neural Networks are used experimentally to model the human 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 supervision of an expert. This calls for a neural network.

Another reason that justifies the use of A.N.N. technology is the ability of A.N.Ns to provide sensor fusion which is the combining of values from several different sensors. Sensor fusion enables the A.N.Ns to learn complex relationships among the individual sensor values, which would otherwise be lost if the values were individually analyzed. In medical modeling and diagnosis, this implies that even though each sensor in a set may be sensitive only to a specific physiological variable, A.N.Ns are capable of detecting complex medical conditions by fusing the data from the individual biomedical sensors.

A Novel Approach to Modelling and Diagnosing the Cardiovascular System [13].

3.5 Instant Physician.

An application developed in the mid-1980s called the "instant physician" trained an auto-associative 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. After training, the net can be presented with input consisting of a set of symptoms; it will then find the full stored pattern that represents the "best" diagnosis and treatment.

3.6 Electronic Noses.

A.N.Ns are used experimentally to implement electronic noses. Electronic noses have several potential applications in telemedicine. Telemedicine is the practice of medicine over long distances via a communication link. The electronic nose would identify odours in the remote surgical environment. These identified odours would then be electronically transmitted to another site where a door generation system would recreate them. Because the sense of smell can be an important sense to the surgeon, telesmell would enhance telepresent surgery.

3.7 Traditional Difficulties in handling Medical Data

Medical data typically requires a large amount of preprocessing in order to be useful. There is numeric and textual data interspersed. Frequently different symbols are used with the same meaning; "male" may be denoted as "M", "m", 0 or a variety of other formats. One medication or condition may be commonly referred to by a variety of names. There is often a redundancy of data; age may appear in several places. Erroneous data is very common; medical terms are frequently misspelled. Finally, medical data is frequently sparse; when a structure is imposed on medical data much of the structure remains empty for a large portion of the population due to the breadth required of any structure.

A robust data preprocessing system is required in order to draw any kind of knowledge from even medium sized medical data sets. The data must not only be cleaned of errors and redundancy but organized in a fashion which makes sense for the problem; in this study's context, the data must be organized so that the benefits of using unsupervised Neural Networks may be maximized.

3.7.1 Organizing Medical Data

Standard techniques were employed to clean erroneous and redundant data; for example, "GI Bleeding", "Gi Bleed" and "Gastrointestinal Bleeding" were all mapped to "Gastrointestinal Bleeding". If "Albuterol" appeared more than once for a patient, multiple entries were discarded.

The data set used contains only data at the leaves; each leaf node is present in only one or two tuples on average. This poses two problems: any conclusions formed would be statistically insignificant and the level of computation required for such an analysis would be exceedingly high due to the existence of roughly ten thousand different leaf nodes. To alleviate these problems the data was processed at the root level of each tree. At the root level each tree collapses into much fewer nodes; fourteen root level drugs, sixteen root level topographies and ten root level morphologies. By constraining all data to the root level the degree of differentiation has been greatly reduced from thousands to 40 (14 + 16 + 10). The trade-off in cost is a great reduction in precision, while a benefit is the possibility of detecting trends within the data at the general level.

As the trees were collapsed to the root level the per tuple data was converted to bipolar format. For every tuple each of the 40 root level nodes was assigned a value of either 1 or -1 depending on whether any data existed for the leaves of that root node. The node is assigned a value of 1 if at least one data value is present at the leaves. The node is assigned a value of –1 if no data is at the leaves. In other words, only existence is preserved; quantity is lost. The end result for each tuple is a 40-dimensional bipolar array. The original data is at most 18-dimensional - containing one to six drugs, one to six topographies and one to six morphologies. Each dimension may contain one of thousands of values, and there is no structure to the data due to variable dimensionality and null and repeated values. By contrast, every tuple of the transformed data contains 40 dimensions, each of which may take one of only two values, 1 or -1. The transformed data is much more consistent and lends itself to computationally intensive analysis such as Neural Networks. NeuralWare, Inc., Data Sculptor Manual. Pittsburgh: NeuralWare Technical Publications Group, 1994 [14].

3.8 Tumor Diagnosis Using Backpropagation Neural Network Method.

For characterization of skin cancer, an artificial neural network (A.N.N.) method has been developed to diagnose normal tissue, benign tumor and melanoma.

The pattern recognition is based on a three-layer neural network fuzzy learning system. In this study, the input neuron data set is the Fourier Transform infrared (FT-IR) spectrum obtained by a new Fiberoptic Evanescent Wave Fourier Transform Infrared (FEW-FTIR) spectroscopy method in the range of 1480 to 1850 cm-1. Ten input features are extracted from the absorbency values in this region. A single hidden layer of neural nodes with sigmoids activation functions clusters the feature space into small subclasses and the output nodes are separated in different nonconvex classes to permit nonlinear discrimination of disease states. The output is classified as three classes: normal tissue, benign tumor and melanoma. The results obtained from the neural network pattern recognition are shown to be consistent with traditional medical diagnosis. Input features have also been extracted from the absorbency spectra using chemical factor analysis. These abstract features or factors are also used in the classification.

3.9 Neural Network for Breast Cancer Diagnosis

Breast cancer diagnosis has been approached by various machine learning techniques for many years. This paper describes two neural network based approaches to breast cancer diagnosis, both of which have displayed good generalisation.

The first approach is based on evolutionary artificial neural networks. In this approach, a feedforward neural network is evolved using an evolutionary programming algorithm. Both the weights and architectures (i.e., connectivity of the network) are evolved in the same evolutionary process. The network may grow as well as shrink.

The second approach is based on neural network ensembles. In this approach, a number of feedforward neural networks are trained simultaneously in order to solve the breast cancer diagnosis problem cooperatively. The basic idea behind using a group of neural networks rather than a monolithic one is divide-and-conquer. The negative correlation training algorithm we used attempts to decompose a problem automatically and then solve them.

3.10 Classify Breast Cancer Cells with Neural Network Software.

Breast cancer cells are traditionally examined under a microscope by a human, who decides the degree of cancer present. People are inconsistent in these judgements from day to day and from person to person.

A neural network that classifies breast cancer cells has been developed. The system was developed by Andrea Dawson, MD of the University of Rochester Medical Center, Richard Austin, MD of the University of California at San Francisco, and David Weinberg, MD, PhD of the Brigham and Womens' Hospital and Harvard Medical School of Boston [15]. Initial comparisons showed that BrainMaker is in good agreement with human observer cancer classifications.

Cancer cells are measured with the CAS-100 (Cell Analysis System, Elmhurst, IL). There are 17 inputs to the neural network which represent morphometric features such as density and texture. There are four network outputs representing nuclear grading. The cancerous nucleus is graded as being well, moderate, or poorly differentiated, or as benign. Correct grade assignments were made between 52% and 89% of the time on cases not seen during training. The lower success rate (for well differentiated) may have been due to the smaller percentage of this type in the training set. In addition, heterogeneity is much lower in well-differentiated tumors. Cancerous nuclei were classified within one grade of the correct grade.

3.11 Diagnose Heart Attacks with Neural Network Software.

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

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

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

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

3.12 A Computer Program for Neural Network Aided Diagnosis of Inherited Metabolic Diseases

We have developed a prototype computer program, MetaNet, that uses a combination of artificial neural networks and knowledge-based expert systems to assist in the diagnosis of inborn errors of metabolism in children.

Results of amino acid analysis data of normal children, and of patients diagnosed with a number of amino acid and organic acid abnormalities were used as inputs to train the neural network component of the program. To diagnose new cases, plasma or urinary amino acid results are entered. The knowledge-based expert system then asks questions of the user regarding the presence or absence of common clinical and/or biochemical abnormalities.

Using both the amino acid data and the answers to the questions, the MetaNet program integrates the output of the neural network and the results of the knowledge-based expert system to yield a provisional diagnosis.

The diagnostic output is accompanied by a numerical *belief vector*, which indicates the degree of confidence of the program in the diagnosis. Altering any of the input variables followed by reprocessing of the data generates a new diagnostic output and a revised belief vector. This allows analysis of the importance of any input variable to the proposed diagnosis. The neural network component consists of eight, three-layer neural networks that are trained using a back-propagation approach. Analysis of the hidden layers following training of the neural network revealed both expected and novel, unexpected connections between specific diagnoses and clusters of amino acids. Such data may be used as a guide for future investigation of the contribution of the metabolism of specific amino acids to amino acid disorders.

3.13 Using Artificial Neural Networks To Diagnose Coronary Artery Disease.

Today's current technology for imaging of coronary artery disease, Single Photon Emission Computed Tomography (SPECT) operates by collecting a series of two dimensional scintigraphic images from around the body. In each image, a pixel's value is the count of the number of photons that were recorded by the gamma camera in that spot. From these images a three dimensional model of the chest is created, and this model is subjected to an algorithm which produces a two dimensional polar plot of the regions of the heart which the physician uses as an aid in diagnosing CAD.as in figure 3.1.

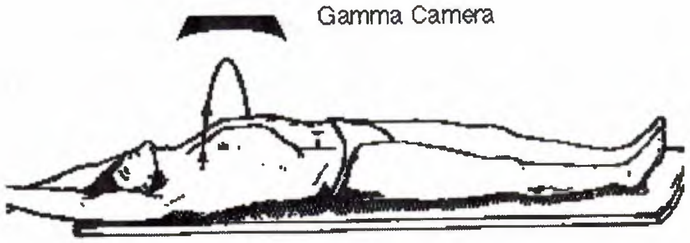


Figure 3.1 shows how to plot a three dimensional image of the heart.

However, this technique is far from perfect. Therefore, a new technology based on neural networks was developed to evaluate the usefulness of A.N.N.s to diagnose CAD. This new technology utilizes the digital format of the SPECT data and the power of computers to manipulate that data.

Using the initial two dimensional images of the chest from the SPECT, an algorithm was developed that extracted the region around the heart in each image. These subimages were then reduced and fed as the input into a neural network. The network was trained on 31 patients, 16 healthy and 15 suffering from CAD. A disjoint set of patients, 5 each of healthy and diseased, was used as a validation set to determine when to halt the training. After the training was stopped, the network correctly classified 14 of the 15 diseased patients in the training set, and 14 of the 16 healthy patients. Of the patients in the validation set, it classified all of the patients correctly. This network was then presented another disjoint set of patients. Of the 7 diseased patients in this set, 4 were classified correctly, while 4 of the 6 healthy patients in this set were classified correctly.

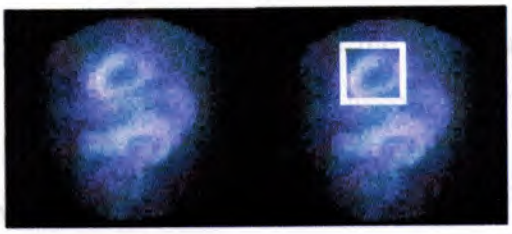


Figure 3.2 SPECT images indicating heart location.

In investigating these results, it was discovered that almost all of the disease patients in the training and validation sets suffered from disease in the inferolateral wall of the heart. Of the 7 patients in the final testing set, the 4 correctly classified patients suffered from inferolateral disease, while the other three had disease in other coronary zones. Therefore, the network was trained to identify only disease in this inferolateral wall of the heart, while patients with disease not in this region would cause confusion for the A.N.N. Since the A.N.N. was not presented patients with disease in other coronary zones, the network could not learn to diagnose disease in these areas since it learns by example.

3.13.1 Results.

This A.N.N. was able to distinguish between patients suffering from CAD in the inferolateral wall and patients who did not suffer from disease in this region. This implies that it is indeed feasible to train an A.N.N. that can diagnose CAD.

One lesson learned is that reducing the size of the data before training the network is crucial to reduce the degrees of freedom in the network. Another issue of concern is how the diagnoses for each of these patients is acquired, since the A.N.N. needs this information to learn how to classify the patients.

The lessons learned in this project have been spun into a series of projects loosely labeled as RADANN - Radiology Diagnostics using Artificial Neural Networks. This family of projects consists of diagnosing various conditions of the human body using radiological images as inputs to A.N.N. based diagnostic tools. David D. Turner [16].

3.14 Summary

In this chapter which is specialized in a Neural Network in medicine. No one denies that Neural Network has played a very essential rule in medicine as well as in many other applications. So here N.N. in order for the physician to hold or to deal with large quantities of data, the N.Ns has been placed instead of him, so it can handle these quantities and other samples of data and then it will not be that difficult to diagnosis the diseases and tumors as will.

Neural Networks also gives facilities to physicians to access the record of the past patients data, which that may help the hospitals to minimize the excessive costs from unnecessary laboratory testing and ineffective patient treatment. N.N. also has been used to develop the drugs for treating the cancer and other tumors.

CHAPTER FOUR

SEGMENTATION OF MEDICAL IMAGES

4.1 Introduction.

The general objective of segmentation of medical images is to find regions which represent single anatomical structures. As in the case of natural-scene vision, segmentation is a crucial step in building systems for the automatic analysis of the imaged ``world". In fact, the availability of regions which represent single structures makes tasks such as interactive visualization and automatic measurement of clinical parameters directly feasible. In addition, segmented images can be further processed with computer vision techniques to perform higher-level tasks such as shape analysis and comparison, recognition and clinical decision-making. D.H. Ballard and C.M. Brown. *Computer Vision*. Prentice-Hall, Englewood Cliff, NJ, 1982 [17].

Unfortunately, segmenting medical images is a very difficult task. This is due to noise, masking structures, biological shape variability, tissue in homogeneity, imaging-chain anisotropy and variability, etc. In order to overcome these problems most researchers have adopted the strategy of exploiting different kinds of *a priori* information about the imaged structures. However, up to now, segmentation systems based on conventional algorithmic techniques or on symbolic knowledge-representation and processing have often shown a limited robustness and, in most cases, have required considerable efforts for eliciting knowledge.

Recently, Artificial Neural Networks A.N.Ns have partially overcome these drawbacks. The main advantages of A.N.Ns are capability to learn from examples and to generalize what has been learned (feed-forward nets), noise rejection, fault tolerance, optimum-seeking behavior (recurrent nets).

In this paper we describe three architectures for medical-image segmentation based on ANNs. These architectures, despite their different approaches, show that ANNs, can exploit and integrate different kinds of *a priori* information contained in medical images. Such information is related to:

a) The imaged district and the structure(s) of interest inside it (anatomical knowledge).

b) The characteristics of the adopted imaging modality (physical knowledge).

c) ``Regularities" of biological structures.

4.2 Segmentation Based on Anatomical Knowledge.

A first issue to be addressed in exploiting anatomical knowledge for medical-image segmentation is the definition of adequate models representing the typical appearance of the structures being considered. Such models are used to produce *a priori* expectations about the input data. Consequently, a second crucial issue in the use of anatomical knowledge is the correct integration of actual input data with model-based expectations.

Segmentation systems based on symbolic models have been proposed in past years by several authors Sai Prasad Raya. Low-level segmentation of 3-D magnetic resonance brain images - a rule-based system. *IEEE Transactions on Medical Imaging* [18]. However, in general, those systems have not obtained the expected results. This is mainly due to the intrinsic complexity and variability of biological objects which heavily hamper the elicitation of knowledge from expert observers. In addition, with such approaches, the correct integration between *a priori* knowledge and input data is usually difficult to achieve.

Feed-forward ANNs can naturally integrate anatomical knowledge with the information contained in the images without requiring the formulation of explicit object-descriptions. In fact, the output of a trained neural network relies both on the input data and on the *a priori* expectations that have been stored in the network connection during the learning phase. Thanks to this property ANNs can effectively face the problems encountered in knowledge-based segmentation of medical images.

On the basis of these ideas we have developed a system (based on feed-forward networks trained with the back-propagation algorithm) for the segmentation of target structures in tomographic images, and of low-contrast lesions in standard projection radiography. The structure of the system is described in the next subsection.

4.2.1 The System.

The system consists of a set of basic modules (one for each structure to be segmented) such as the one schematized in Figure 4.1. Each module includes three major blocks: a retina, an Attention Focuser (AF) and a Region Finder (RF). The retina is the input section of the system. It preprocesses the input image to produce a low-resolution output picture, utilized by AF to locate the desired structure, and a high-resolution output picture, used for segmentation by RF.

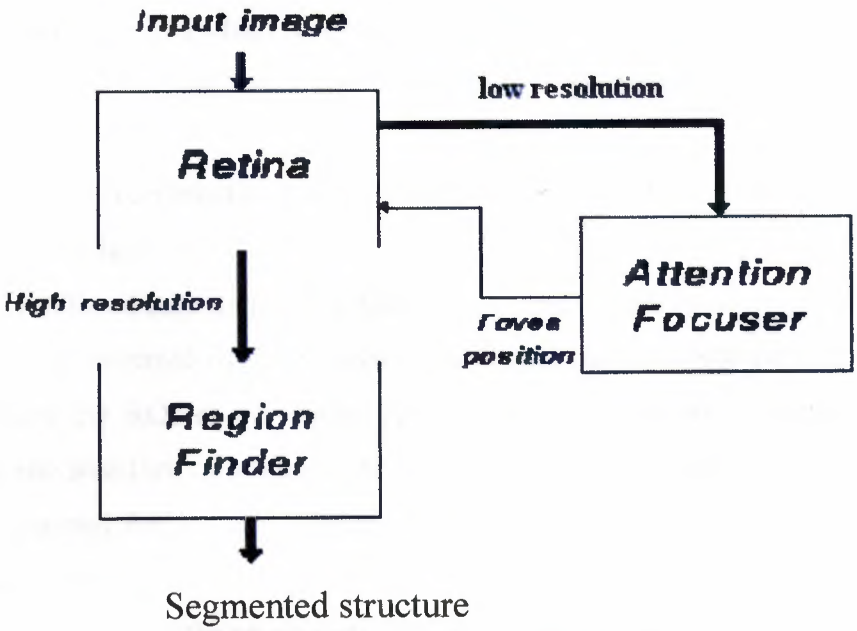


Figure 4.1 Basic module architecture.

The retina is composed of an input layer including as many neurons as image pixels, and an output layer with a reduced number of neurons. As in biological retinas, the connections between the input and the output neurons are local and are arranged in overlapping receptive fields centered on each output neuron. We use receptive fields with Gaussian weights for MR and coronary trees images CT and with Laplacian of Gaussian (LoG) weights for X-ray images. LoG-weighted receptive fields filter out the low-frequency components of X-ray images which are responsible for background variability. The retina also includes a moving region, called the fovea, which performs the same operation at a higher spatial resolution on a Region of Interest (ROI) selected by AF.

As to AF, we have designed two different structures. When a single entity has to be segmented, AF is a fully connected network with:

a) As many input neurons as the number of pixels of the low-resolution image produced by the retina.

b) Two hidden layers.

c) An output layer which encodes the coordinates of the centroid of the structure under consideration.

When multiple structures are of interest, AF has:

a) An input layer arranged as a square mask which scans the image.

b) Two hidden layers.

c) One output neuron which fires when a target structure enters the input mask.

It is worth noting that the use of an attention-focusing mechanism has two main advantages:

a) The overall computation load is reduced (only the ROI is processed at a high spatial resolution),

b) The produced segmentation is insensitive to translation.

Once the centroid of the structure considered has been computed by AF, the fovea processes the ROI at the higher spatial resolution and RF extracts the pixels belonging to the structure of interest. The topology of RF is depicted in Figure 4.2. RF is a block-connected five-layer net which operates like a (nonlinear space-varying) filter by processing, for each pixel:

a) The gray level of the pixels contained in a square mask.

b) The position of the mask (given by an appropriate encoding of the coordinates of its central pixel).

As shown in Figure 4.2, these different kinds of information are processed independently by two different fully connected sub-networks joined in a common layer. This network topology provides an optimal integration of input data with *a priori* knowledge. The activation of the output unit indicates whether the processed pixel belongs to the considered organ.

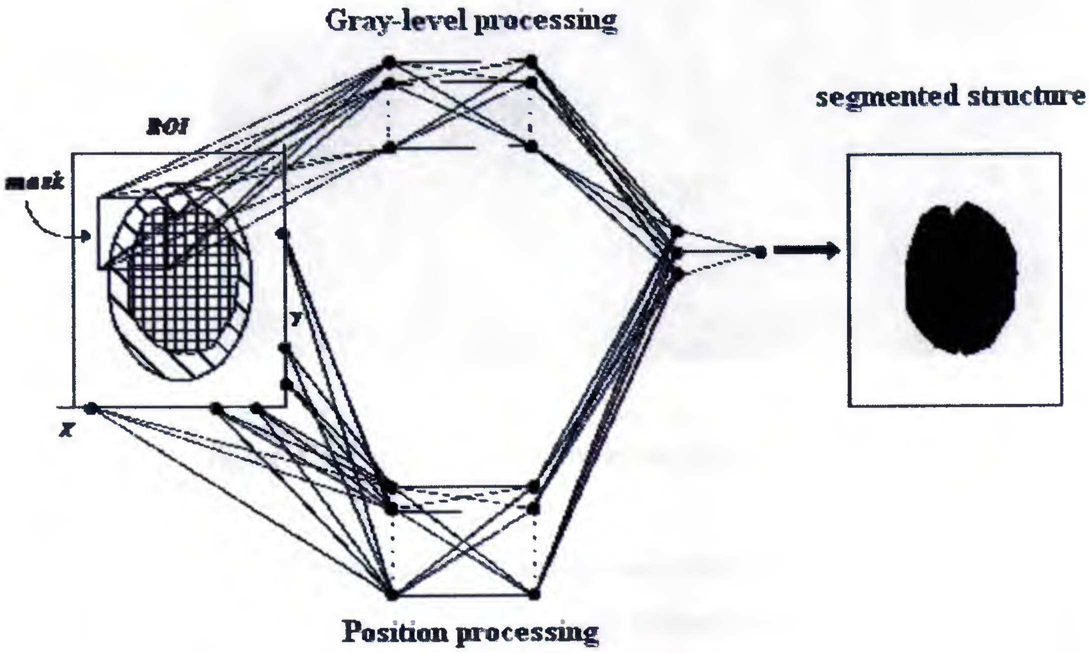


Figure 4.2 Region finder topology

4.2.2 Results

The described architecture has been applied to the segmentation of the brain in MR images and the spinal chord in CT images. The system has also been employed in segmenting low-contrast lesions in standard chest radiographs. In each case, the system has been trained and tested both on computer simulated phantoms and on clinical images.

In the experiments with tomographic images (represented by 256×256 gray-level matrices), the retina produces 16×16 Gaussian filtered low-resolution images. AF has been implemented as a 16×16+32+32+32 network (each coordinate being encoded by 16 output units).

In Figure 4.3 we show a sequence of MR tomograms of the head: the contours of the brain as segmented by the system are superimposed on the original images. The system has shown remarkable sensitivity and specificity, the rate of correct pixel-classifications being above 95%.

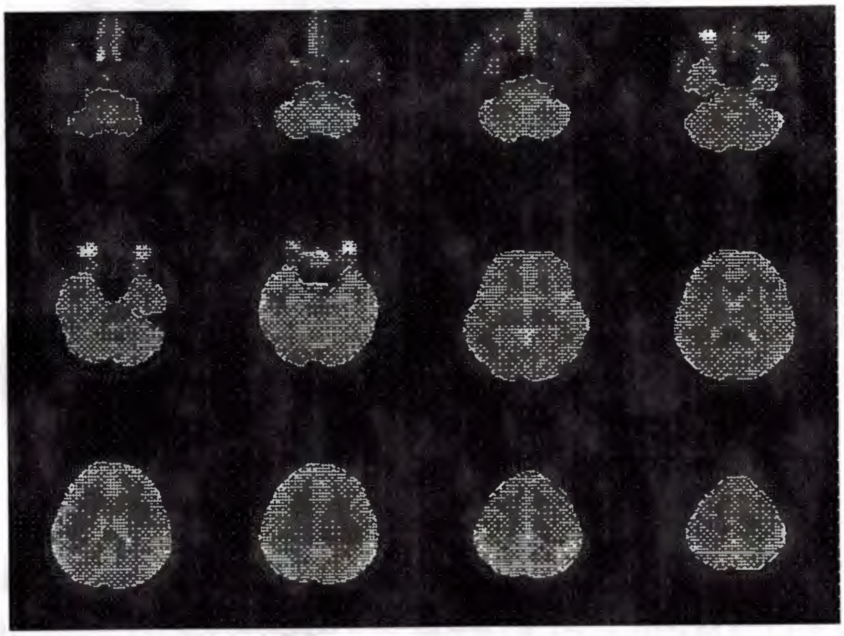


figure 4.3 Segmentation of the brain from an MR image sequence.

In the experiments with chest radiographs (represented by 768×768 gray-level matrices), the retina produces 256×256 LoG filtered low-resolution images. AF has been implemented as a 19×19+32+32+1 network which scans the low-resolution picture. In this way, a set of points (*attention foci*) which indicate possible lesions are produced. RF analyzes the ROIs around the *attention foci* and, for each of them, produces a binary output which represents the segmentation of nodule-like patterns. The segmentation can then be analyzed by a neural recognition system (a three-layer fully-connected network), which labels each region as pathological or normal. In Figure 4.4 we illustrate the typical operation of this system. The four panels show (top to bottom, left to right): a radiogram with a malignant nodule encircled by the radiologist, the *attention foci* produced by AF, the regions segmented by RF and the structures classified as pathological. Experimental results indicate good sensitivity and reasonable specificity in detection of parenchyma lesions. As to the segmentation phase, the rate of correct pixel-classifications has been above 90%.

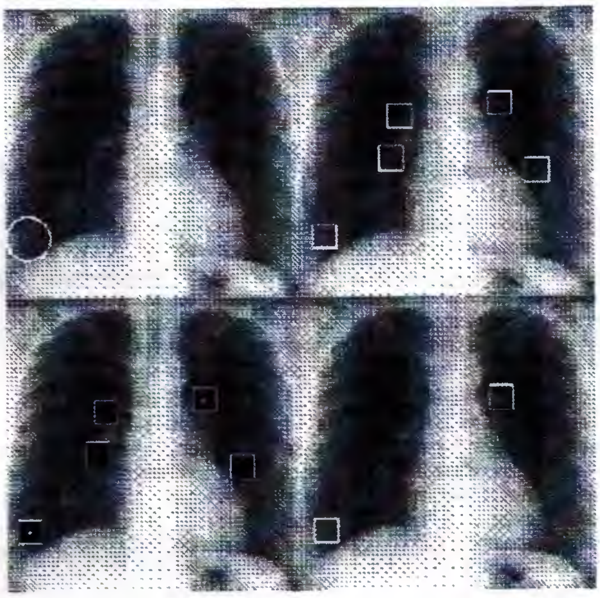


figure 4.4 Steps of the segmentation of lung nodules

4.3 Computerized Detection of Pulmonary Nodules.

Currently, radiologists can fail to detect lung nodules in up to 30% of actually positive cases. If a computerized scheme could alert the radiologist to locations of suspected nodules, then potentially the number of missed nodules could be reduced. We are developing such a computerized scheme that involves a difference-image approach and various feature-extraction techniques.

4.3.1 Method.

The method involves producing two processed digital images from one chest radiograph: one in which the signal of the nodule is enhanced and the other in which it is suppressed. Both linear and nonlinear filtering operations are used. The difference between the two processed images yields an image of the signal superimposed on a simplified background.

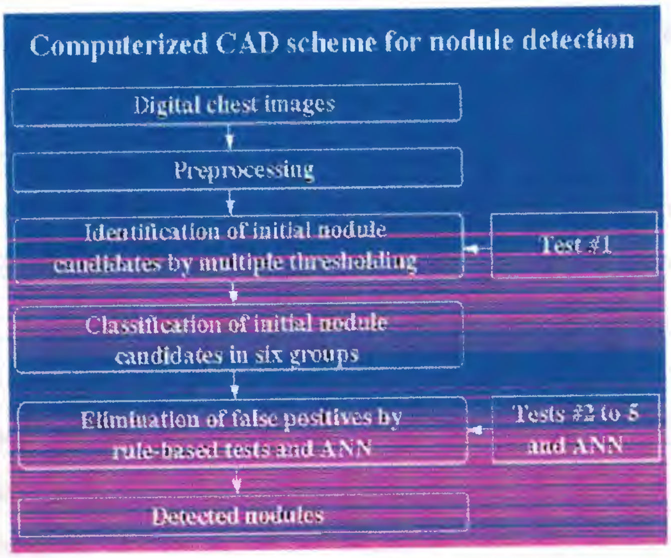
Feature-extraction techniques are then applied to the difference image to distinguish nodules from normal anatomic background patterns

4.3.2 Results.

The computerized detection scheme was used in the evaluation of posteroanterior chest radiographs from 120 clinical cases that included nodules of various subtlety and sizes (5-30 mm). The presence and location of the nodules were verified by means of computed tomography or follow-up radiography.

The computer aided diagnosis (CAD) program achieved a true-positive detection rate of approximately 75% with an average of approximately one false-positive detection per chest image as described in figure 4.5. Computer outputs indicating locations of potential lesions are marked by arrows on the chest images figure 4.6.

Observer performance studies indicate that ROC curves of radiologists for detection of lung nodules were significantly improved when the computer output was available as an aid.



Computerized CAD scheme for nodule detection

Digital chest images

Preprocessing

Identification of initial nodule candidates by multiple thresholding

Test #1

Classification of initial nodule candidates in six groups

Elimination of false positives by rule-based tests and ANN

Tests #2 to 5 and ANN

Detected nodules

Figure 4.5 Flowchart of the CAD scheme.

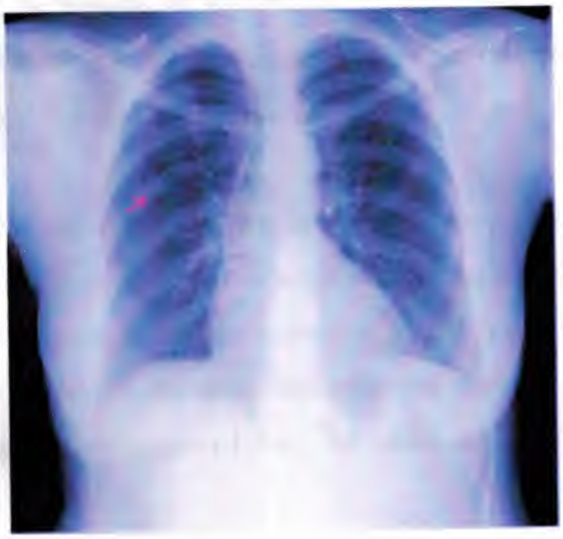


Figure 4.6 Output of the scheme, with an arrow showing a suspicious area (in this case, a true nodule.)

4.4 Segmentation Based on the Physics of Image Generation.

Some imaging techniques, such as MR, allow for reproducible measurements of parameters which characterize different tissues. Such measurements can be analyzed with classification algorithms in order to obtain segmentations.

As regards physical parameters which characterize MR spin echo sequences, the dependence of signal intensity I on tissue properties and sequence parameters can be described by the following approximate equation: $I = k(\rho, T_1, TR)\, e^{-\frac{TE}{T_2}}$ where ρ (proton density), T_1 and T_2 (relaxation times) are tissue-dependent parameters, while TE (echo time) and TR (repetition time) are parameters of the acquisition sequence that can be set by the operator. This equation shows that the characteristics of spin echo images strongly depend on T_2 and contrast between different tissues can be adjusted with a suitable choice of TE. Given a set of MR images of the same slice acquired with different TEs (multiecho sequence), T_2 can be estimated for each pixel, thus allowing for T_2 -based tissue classification.

Several authors have reported encouraging results obtained on the segmentation of MR images of the brain using statistical, fuzzy, and neural-network approaches based on this idea. However, a critical problem to be tackled is the presence of different tissues of similar appearance, which hampers classification based only on MR physics. In the case of brain tissues, misclassifications mainly affect subcutaneous fat and white matter, as reported in D. Piraino, S. Sundar, B. Richmond, J. Schils, and J. Thome. Segmentation of magnetic resonance images using a backpropagation neural network. In *Proc. Annual Int. Conf. of IEEE EMBS*, pages 1466-1467, 1991 [19]. and as we verified in our early experiments.

This problem can be solved by integrating knowledge about the physical principles of the imaging device with anatomical knowledge about shape and location of the brain parenchyma. Indeed, if a pixel is known to belong to the brain, no ambiguity can arise between fat and white matter. In the next subsection we describe a neural system for the segmentation of MR spin echo images of the brain which is based on this idea.

4.4.1 The System

The system includes three main modules (see Figure.4.7) which accomplish the following main computational processes:

a) enhancement and analysis of the information provided by signal decay over time.

b) detection of the brain parenchyma.

c) pixel classification into five predefined tissue groups.

The first two modules are based on feed-forward networks trained with the back-propagation algorithm, while the neural classifier which performs the actual segmentation is implemented as a Kohonen topology-preserving map.

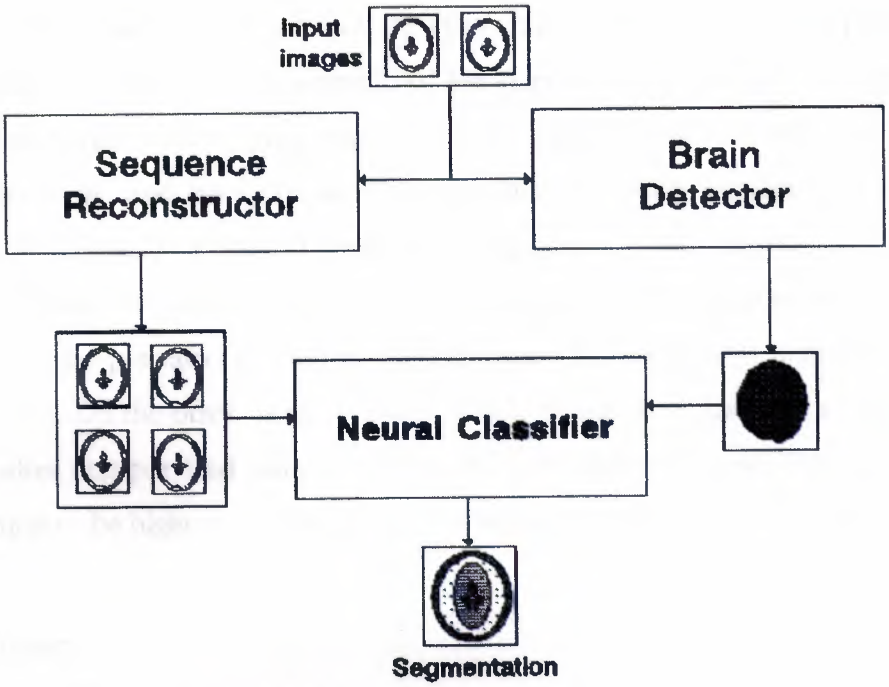


Figure 4.7 Overall structure of the system

The introduction of the first module, that we have called Sequence Reconstructor (SR) enhances information about signal decay and improves the signal-to-noise ratio, in order to crispen the separation among different tissues. SR is a neural model of the whole procedure which, from the acquisition of the raw signal, leads to the generation of the actual MR images. As a result, it synthesizes long-TE images which, along with the input images, generate a four-echo multiecho sequence. Thanks to noise-rejection properties of ANNs, this sequence is considerably less noisy than the corresponding true multiecho sequence. The neural network that performs such a task consists of four layers with 18+6+6+3 units. The inputs are the gray levels of two corresponding 3×3 windows taken from two short-TE images of the same slice. The outputs represent the estimated intensity values of the central pixel of such windows in three images with prefixed longer TEs. Therefore, by successively processing pairs of 3×3 windows centered on each pixel of the input images, the network synthesizes whole long-TE images.

The second module, termed Brain Detector (BD), supplies the neural classifier with *a priori* anatomical knowledge. The architecture of BD is the one described in Figure 4.1 BD has been trained to produce an image in which pixels belonging to brain parenchyma are enhanced.

The outputs of SR and BD are eventually processed by the neural classifier. This module performs the segmentation of the sequence by assigning each pixel to one of the following classes: gray matter, white matter, cerebrospinal fluid, skin or subcutaneous fat and background. The classifier has been implemented as a one-dimensional 256-unit Kohonen self-organizing map. With respect to (standard) backpropagation, this neural paradigm has the advantage of faster training and higher flexibility in the presence of classes characterized by multi-modal distributions in the pattern space. On the other hand, to use a Kohonen map as a classifier, a further step is required after unsupervised training. This step labels each neuron according to the class which contains the highest number of patterns that activate that neuron.

4.4.2 Results

The performances of the system were assessed on a test set of 1250 pixel sequences, extracted from 25 MR slices and classified by an expert radiologist. The test resulted in a global accuracy of 94%. The classifier's best performance (99%) was obtained (thanks to the anatomical information) on skin and subcutaneous fat, while the worst (90%) was obtained on cerebrospinal fluid.

Not surprisingly, suppressing the input produced by BD resulted in a remarkable worsening of the performances, as the accuracy was hardly over 70%in the case of patterns representing skin and subcutaneous fat.

The results that can be obtained by the system are illustrated in Figure 4.8, in which the two input images are shown in the top row, two long-***TE*** images produced by SR in the mid row, and the output of BD and the final segmentation in the bottom row.

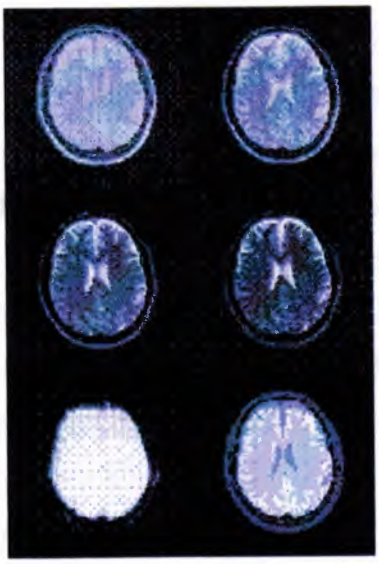


Figure 4.8 The input images (top row), the output of SR(mid row), the output of BD(bottom left), and the final segmentation (bottom right).

4.5 Segmentation Based on Perceptual Principles.

Biological vision is ruled by principles such as perceptual grouping, selection, discrimination, etc. which mostly depend on regularities of nature such as cohesiveness of matter or existence of bounding surfaces. These properties are certainly valid also for the anatomical structures contained in medical images, and can be exploited to build general-purpose segmentation systems for that kind of image.

From the perceptual standpoint, the optimum segmentation algorithm for medical images should be sensitive to small structures and to structures with a low contrast (high discriminating power), and robust with respect to noise, texture and slow intensity-changes (high grouping power).

It stands to reason that these requirements counteract each other and that a trade-off solution is necessary. A first step towards the best compromise between sensitivity and robustness is the definition of a quantitative criterion of goodness of segmentation. Afterwards, this criterion has to be maximized by the segmentation procedure for any specific image. In this view, the problem of medical image segmentation can be considered as a problem of optimization. Unfortunately, for any given image there is a huge number of possible segmentations.

4.6 Summary

In this chapter I have described three different approaches to the segmentation of medical images. Each one of them overcomes the problems usually encountered in such a task by exploring a different kind of priori information. In the first two, trained neural networks have easily been able to extract and subsequently use knowledge about the anatomy of the imaged districts and about the physics of image generation. In the third approach, Hopfield Neural Networks have allowed for a neural implementation of perceptual principles of grouping and discrimination.

Generally, what does the segmentation mean?

The segmentation of medical images is to find the regions which represent single anatomical structures. Means that the medical images are to be divided into small parts, in other word into pixels and then compare each of these pixels with the pixels of other original image pixels in order to differentiate between them and detect the part of these images which belong to the infected organ.

CONCLUSION

During the preparation of this project and searching knowledge has been aquired regarding the latest development of neural network technology. Here Artificial Neural Network has proved how effective it is in our daily life, so A.N.N. has taken the most difficult problem on its own shoulder, it is so effective and accurate and has many advantages and benefits in many branches and it solved a lot of things seem to be easy, but actually it is not, from diagnostic analysis, image analysis to genetic improvement. All of these things can be achieved by using Artificial Neural Network and developing the expert systems and the intelligent systems as well.

Artificial Neural Network is being developed all the time especially in medicine and genetic development. So, from my point of view I think that A.N.N. will come up with new genetic properties to get rid of most dangerous diseases on the humanitarian life.

Chapter one described the history of the Artificial Neural Networks and described the advantage, disadvantage and the problems of the Artificial Neural Networks.

Chapter two described the structure of the A.N.Ns, ways of training the A.N.Ns, learning methods and other applications that involved in, beside the next development in A.N.Ns.

Chapter three have described the use of A.N.Ns in medicine whether in medical diagnostic aids, biochemical analysis. Also described the developing the drugs using A.N.Ns.

Chapter four described three kinds of segmentation that Artificial Neural Networks have used in medicine.

REFERENCES

[1]. Academic writing. From the world wide web: "http://www.dacs.dtic.mil/techs/neural/neural_ToC.html".

[2]. Academic writing. From the world wide web: "http://www.dacs.dtic.mil/techs/neural/neural_ToC.html".

[3]. Academic writing. From the world wide web: "http://www.usps.gov/websites/".

[4]. Academic writing. From the world wide web: "ftp://ftp.cs.cmu.edu/user/ai/pubs/news/comp.ai.neural-nets"

[5]. Academic writing. From the world wide web: "http://www.dacs.dtic.mil/techs/neural/neural6.html"

[6]. Academic writing. From the world wide web: "http://www.pnl.gov/".

[7]. Academic writing. From the world wide web: "http://www.dacs.dtic.mil/techs/neural/neural5.html".

[8]. Academic writing. From the world wide web: "http://www.dacs.dtic.mil/techs/neural/neural6.html"

[9]. Hutchison & Stephens, 1987. From the world wide web: "http://www.singnet.com.sg/~midaz".

[10]. Academic writing. From the world wide web: "ftp://ftp.sas.com/pub/neural/FAQ.html".

[11]. Academic writing. From the world wide web: "http://rodin.wustl.edu/center.html".

[12]. Academic writing. From the world wide web: " http://www.pnl.gov/medical/".

[13]. Academic writing. From the world wide web: "http://www.emsl.pnl.gov:2080/".

[14]. Writing papers. From the world wide web: "http://www.ee.umd.edu/medlab/papers/".

[15]. Academic writing. From the world wide web: "http://www.calsci.com/".

[16]. Academic writing. From the world wide web: "http://www.battelle.org/medical/".

[17]. Academic writing. From the world wide web: " http://www.cs.bham.ac.uk".

[18]. Academic writing. From the world wide web: "http://www.cs.bham.ac.uk".

[19]. Academic writing. From the world wide web: "http://www.cs.bham.ac.uk".