NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

NEURAL NETWORK APPLICATION IN MEDICINE

Graduation Project
COM- 400

Student: MAZHAR GHUNEIM

Supervisor: Assoc.Prof.Dr.Adnan Khashman

Nicosia - 2004

ACKNOWLEDGEMENT

I would like first and foremost to thank Allah whom its accomplishment would not have been possible.

I would like to send my regards to my parents and like also to deeply thank them. I would like to thank my father who always encourages me to achieve my goals. I would like to thank my mother who always prays to me. So I want to present this graduation to my parents, my brothers and sisters whom also helped me to reach this goal so they are the main reason of this successful.

Need less to say this project could not have been completed without the kind assistance and 2nd academic guidance of my supervisor, Assoc.Prof .Dr Adnan Khashman to whom I am greatly thankful.

I am very grateful to my university, and engineering department, the family which accepted me in 2000. I express my gratitude to all university, faculty and staff with whom I have a good fortune

Finally, please let me send a full thankful to all my friends that spent this year with me.I want to send a special regards to my close friends: Fadi Alia, Ahmad Shahin, samer kattab, Ali Abu Khrma, Mohammad Sharaf, Rabie Ahmad, Ahmad Osman and Ahmed Al Farra.

ABSTRACT

Artificial neural networks have been used in a number of different ways in medicine and medically related fields. This project using neural networks application in medicine. This involved the use to solve optimization and dynamical control problems. A general framework for artificial neural networks models is introduced first. Then the main feed-forward and feedback models are presented a number of theoretical and practical aspects of the application of neural networks are presented in this project. Firstly the biological neuron is presented and utilized to try and determine the relationship between the artificial neural network and human secondly, conditions relating to network learning by supervised and unsupervised are examined, with the finding that both the number of hidden nodes, network architecture and the initial conditions of the network are important in determining if a neural network will learn a particular problem.

The next involves attempting to concept a diagnosis prediction via an artificial neural network knowledge base. The aim of this approach the major application of medical information has been the metabolic disease diagnosis. Unfortunately this aim is not realized within the realms of this project, due to problems in training the neural networks, the final aim involves a much more complex system, medical diagnostic aides. This acts as further confirmation of the limitations in the neural network in medicine.

TABLE OF CONTENTS

INTRODUCTION

1. Overview

Let us first summarize the most important features of the neural networks found in the brain. Firstly the brain contains many billions of very special kinds of cell - these are the nerve cells or neurons. These cells are organized into a very complicated intercommunicating network. Typically each neuron is physically connected to tens of thousands of others. Using these connections neurons can pass electrical signals between each other. These connections are not merely on or off - the connections have varying strength which allows the influence of a given neuron on one of its neighbors to be either very strong, very weak (perhaps even no influence) or anything in between. Furthermore, many aspects of brain function, particularly the learning process, are closely associated with the adjustment of these connection strengths. Brain activity is then represented by particular patterns of firing activity amongst this network of neurons. It is this simultaneous cooperative behavior of very many simple processing units which is at the root of the enormous sophistication and computational power of the brain.

Artificial neural networks are modeled after the brain. They typically consist of many hundreds of simple processing units which are wired together in a complex communication network. Each unit or node is a simplified model of a real neuron which fires (sends off a new signal) if it receives a sufficiently strong input signal from the other nodes to which it is connected. The strength of these connections may be varied in order for the network to perform different tasks corresponding to different patterns of node firing activity. This structure is very different from traditional computers.

2. Description of thesis structure

Chapter one is devoted to artifical neural network introduction of the general methods from neural network theory, and provide a breif history of neural network. It also introduces artifical neurons and how it works.

Chapter two studies the structure of neural network. It introduces architecture of neural networks ,this view feed-forward and leads to feedback neural networks. The last subsection of this chapter introduces the supervised neural network and unsupervised neural networks.

Chapter three is specialized in medical application and describe some fields where we can find the neural network in medicine and introduces artificial neural network based cardiovascular modeling.

Chapter four pulse-coupled neural networks one of the major applications in medicine and image analysis medical systems,how the neural network supporting the imaging proccess in medicine will be described inthis chapter

3.The aim of this project

The following aims and objectives are to met throghout the work presented in this thesis these aims can be summarised as:

1. To retrieve some information about the artifical neural network.

2.To investigate and learn about neural network,the emergence,structure and application in real life.

3. To show on medical application of neural network.

4. In addition examples of succesful implementation of neural medical system will be described.

CHAPTER ONE

NEURAL NETWORKS BACKGROUND

1.1 Overview

This chapter will present an introduction to artificial neural network and provide a brief history of neural networks and describe why use a neural network, in this section which is the introduction of neural network, I am going to explain artificial neurons and how they work and introduce a hint of the of the biological neurons.

1.2 Artificial neural networks

An artificial neural network (ANN) is an information-processing paradigm inspired by the way the densely interconnected, parallel structure of the mammalian brain processes information. Artificial neural networks are collections of mathematical models that emulate some of the observed properties of biological nervous systems and draw on the analogies of adaptive biological learning. The key element of the ANN paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements that are analogous to neurons and are tied together with weighted connections that are analogous to synapses.

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

Although ANNs have been around since the late 1950s [1], it wasn't until the mid-1980s, that algorithms became sophisticated enough for general applications. Today ANNs are being applied to an increasing number of real-world problems of considerable complexity. They are good pattern recognition engines and robust classifiers, with the ability to generalize in making decisions about imprecise input data. They offer ideal solutions to a variety of classification problems such as speech and character and signal recognition, as well as functional prediction and system modeling where the physical processes are not understood or are highly complex. ANNs may also be applied to control problems, where the input variables are measurements used to drive an output actuator, and the network learns the control function. The advantage of ANNs lies in their resilience against distortions in the input data and their ability to learn. They are often good at solving problems that are too complex for conventional technologies (e.g., problems that do not have an algorithmic solution or for which an algorithmic solution is too complex to be found) and are often well suited to problems that people are good at solving, but for which traditional methods are not.

1.3 A Brief History of Neural Network

Neural network simulations appear to be a recent development. However, this field was established before the advent of computers, and has survived at least one major setback several areas. Many important advances have been boosted by the use of inexpensive computer emulations. Following an initial period of enthusiasm, the field survived a period of frustration and disrepute. During this period when funding and professional support was minimal, important advances were made by relatively few researchers. These pioneers were able to develop convincing technology which surpassed the limitations identified by Minsky and Papert. Minsky and Papert, published a book (in 1969) in which they summed up a general feeling of frustration (against neural networks) among researchers, and was thus accepted by most without further analysis. Currently, the neural network field enjoys a resurgence of interest and a corresponding increase in funding.

First Attempts: There were some initial simulations using formal logic. McCulloch and Pitts (1943) [2], developed models of neural networks based on their understanding of neurology. These models made several assumptions about how neurons worked. Their networks were based on simple neurons which were considered to be binary devices with fixed thresholds. Another attempt was by using computer simulations. Two groups (Farley and Clark, 1954; Rochester, Holland, Haibit and Duda, 1956) [3]. The first group IBM researchers maintained closed contact with neuroscientists at McGill University. So whenever their models did not work, they consulted the neuroscientists.

This interaction established a multidisciplinary trend which continues to the present day, but psychologists and engineers also contributed to the progress of neural network simulations. Rosenblatt (1958) stirred considerable interest and activity in the field when he designed and developed the Perceptron. The Perceptron had three layers with the middle layer known as the association layer. This system could learn to connect or associate a given input to a random output unit. Another system was the ADALINE (ADAptive Linear Element) which was developed in 1960[4], by Widrow and Hoff of Stanford University. The ADALINE was an analogue electronic device made from simple components. The method used for learning was different to that of the Perceptron; it employed the Least-Mean-Squares (LMS) learning rule.

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

Klopf (A. Henry Klopf) in 1972 developed a basis for learning in artificial neurons based on a biological principle for neuronal learning called heterostasis. Werbos (Paul Werbos 1974) developed and used the back-propagation learning method, however several years passed before this approach was popularized. Back-propagation nets are probably the most well known and widely applied of the neural networks today. In essence, the back-propagation net. Is a Perceptron with multiple layers, a different threshold function in the artificial neuron, and a more robust and capable learning rule. A Mari (A. Shun-Ichi 1967) was involved with theoretical developments: he published a paper which established a mathematical theory for a learning basis (error-correction method) dealing with adaptive pattern classification. While Fukushima (F. Kunihiko) developed a step wise trained multilayered neural network for interpretation of handwritten characters. The original network was published in 1975 and was called the Cognitron.

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

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

1.4 What is a neural network?

Neural Networks are a different paradigm for computing:

1. von Neumann machines are based on the processing/memory abstraction of human information processing.

2. Neural networks are based on the parallel architecture of animal brains.

Neural networks are a form of multiprocessor computer system, with

1. Simple processing elements

2. A high degree of interconnection

3. Simple scalar messages

4. Adaptive interaction between elements

A biological neuron may have as many as 10,000 different inputs, and may send its output (the presence or absence of a short-duration spike) to many other neurons. Neurons are wired up in a 3-dimensional pattern.

Real brains, however, are orders of magnitude more complex than any artificial neural network so far considered.

1.5 Why Use a Neural Network?

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

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

Self-Organization: An ANN can create its own organization or representation of the information it receives during learning time.

Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.

Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

1.6 Are There Any Limits to Neural Networks?

The major issues of concern today are the scalability problem, testing, verification, and integration of neural network systems into the modern environment. Neural network programs sometimes become unstable when applied to larger problems. The defense, nuclear and space industries are concerned about the issue of testing and verification. The mathematical theories used to guarantee the performance of an applied neural network are still under development. The solution for the time being may be to train and test these intelligent systems much as we do for humans. Also there are some more practical problems like: The operational problem encountered when attempting to simulate the parallelism of neural networks. Since the majority of neural networks are simulated on sequential machines, giving rise to a very rapid increase in processing time requirements as size of the problem expands. Solution: implement neural networks directly in hardware, but these need a lot of development still. Instability to explain any results that they obtain. Networks function as "black boxes" whose rules of operation are completely unknown.

1.7 How the Human Brain Learns?

Much is still unknown about how the brain trains itself to process information, so theories abound. In the human brain, a typical neuron collects signals from others through a host of fine structures called dendrites. The neuron sends out spikes of electrical activity through a long, thin stand known as an axon as in figure 1.2, which splits into thousands of branches. At the end of each branch, a structure called a synapse converts the activity from the axon into electrical effects that inhibit or excite activity from the axon into electrical effects that inhibit or excite activity in the connected neurons as in figure 1.3. When a neuron receives excitatory input that is sufficiently large compared with its inhibitory input, it sends a spike of electrical activity down its axon. Learning occurs by changing the effectiveness of the synapses so that the influence of one neuron on another changes

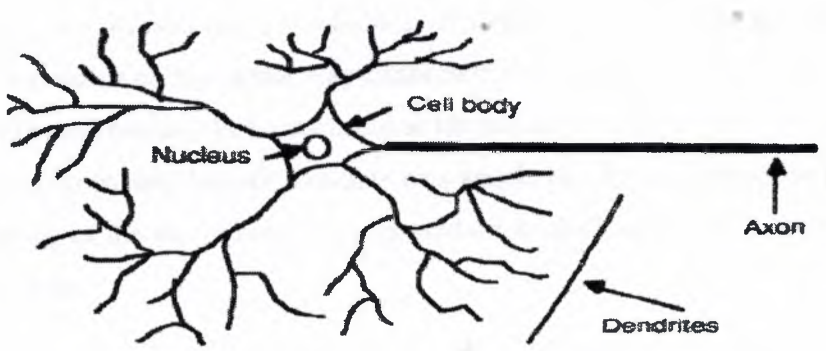

Figure 1.1. Components of a neuron

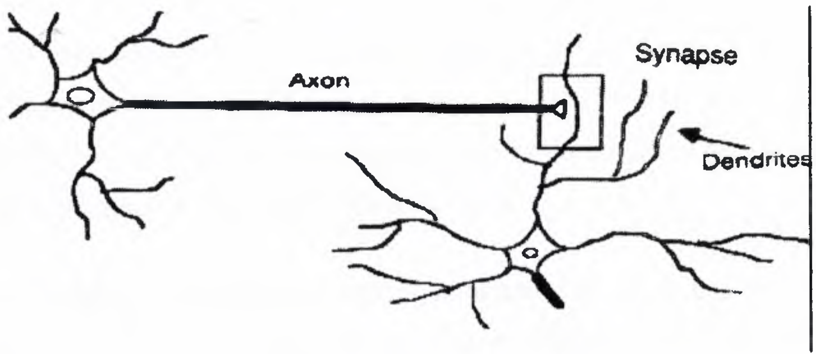

Figure 1.2. The synapse

1.8 Where are neural networks going?

A great deal of research is going on in neural networks worldwide. This ranges from basic research into new and more efficient learning algorithms, to networks which can respond to temporally varying patterns (both ongoing at Stirling), to techniques for implementing neural networks directly in silicon. Already one chip commercially available exists, but it does not include adaptation. Edinburgh University has implemented a neural network chip, and is working on the learning problem.

Production of a learning chip would allow the application of this technology to a whole range of problems where the price of a PC and software cannot be justified.

There is particular interest in sensory and sensing applications: nets which learn to interpret real-world sensors and learn about their environment.

1.9 Artificial Neurons And How They Work?

A neural net is a physical (as in electronics) or virtual (a computer program) collection of nodes or neurons each in some way connected to the other. Each neuron has several inputs and several outputs. Input starts out as the message from an array of sensors. This message is often passed through associate nets which, in a vision system, do a lot of the pre processing of the signal before it is passed on to neurons based on the McCulloch and Pitt's model.

Warren McCulloch and Walter Pitts were a team of neurophysiologist and logician who in 1943 built a model involving resistors and amplifiers which mimicked what was known about natural neurons. Neurons take weighted inputs and then, depending on the result, either fire or do not. This firing is then passed onto several other neurons which take this input and, according to the weighting, act or do not act. The whole model is a network of interconnected cells, each affecting the next.

Eventually the signal in a neural net reaches an output stage. This can be a value (male, female) or an array of output (sound or a picture). At first this result will be near random until the net has been trained, and trained correctly. The net has to receive enough information through the input to be able to make the correct assumptions. For example, one neural net was being used by the military to aid the recognition of tanks. A net was given different pictures of tanks and had to decide whether they were Russian or American. Each time the net got it wrong the net would learn and reorganize its connections and weights. Eventually the net was achieving perfect results. Other pictures from the same set as the training photographs where also correctly sorting into American and Russian. Problems arose when a new set of photos of the same tanks was given to the machine. This time it went back to making mistakes. This was puzzling until someone pointed out the times of the days the photos were taken. The shadows on the trees and tanks fell at a different angle on the American photos and the Russian photos so the computer was sorting the photos by time of day and not by shape of tank. After different sets of photos were used at varying times of day the net learnt the error of its ways and went back to being correct most of the time.

This could be said to be an example of the old adage, garbage in, garbage out. In this case it shows that for a correct assumption to be reached by man or machine all the correct information must be available. If there are inconsistencies or we cannot work out why we get an answer we put it down to common sense. This is the same as neural nets. The weighting of neurons leads to the guessing of answers some of the time, using information known to fill in the gaps.

In the future computers may be a hybrid of Neural net and conventional Turing based computing. Conventional computing has the advantage of being logical and fast in known mathematical problems. Neural nets are not good at number crunching, much as the human brain finds sums harder to handle that music. Instead they excel in pattern recognition, in tasks that require filtering and analyzing data.

What should be pointed out is that current neural networks are about as intelligent as a stupid insect. Neural computing, despite its history, is still a young subject and yet has to be fully understood with true precision. Having said that it has already produced complex results and is being used in many different fields. In the future HAL type computers could be totally possible. Our emotions are our motivation for the things we do; with an artificial intelligence these motivations might be totally different. As with the rest of a life, there is no reason why we our creations should have to take natures.

Future man made intelligences may live their entire lives in environments alien to the human mind. They could exist in different bodies and spend their time thinking about things we would deem unimportant. They could be specialist intelligences possibly not directly comparable to our intelligence. For an example of how different a possible intelligence can be you only need look at the second closest intelligence on the planet, dolphins. The dolphins' and whales' worlds are radically different; it is a world of the oceans. We now know that dolphins and whales have a symbolic language, if a lot simpler than human language. Dolphins give each other name but, like whales, spend time on navigation as we spend on trying to manipulate tools. Different environments put different priorities on a creature living in it; it is currently hard for us to imagine what form of intelligence a creature living purely in the data sphere would take.

1.10 The Biological Neuron

The brain is a collection of about 10 billion interconnected neurons. Each neuron is a cell that uses biochemical reactions to receive process and transmit information. Figure 1.4 shows schematic of biological neuron.

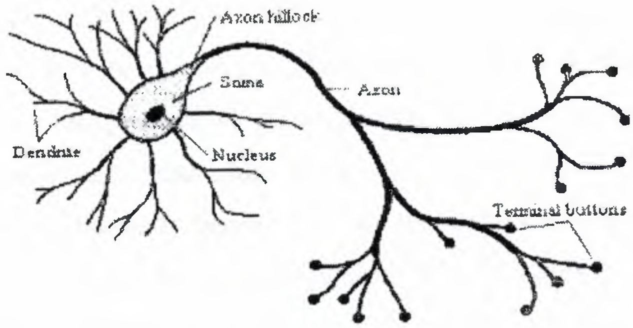

Figure 1.3. Schematic of biological neuron

A neuron's dendrites tree is connected to a thousand neighboring neurons. When one of those neurons fire, a positive or negative charge is received by one of the dendrites. The strengths of all the received charges are added together through the processes of spatial and temporal summation. Spatial summation occurs when several weak signals are converted into a single large one, while temporal summation converts a rapid series of weak pulses from one source into one large signal. The aggregate input is then passed to the soma (cell body). The soma and the enclosed nucleus don't play a significant role in the processing of incoming and outgoing data. Their primary function is to perform the continuous maintenance required to keep the neuron functional. The part of the soma that does concern itself with the signal is the axon hillock. If the aggregate input is greater than the axon hillock's threshold value, then the neuron fires, and an output signal is transmitted down the axon. The strength of the output is constant, regardless of whether the input was just above the threshold, or a hundred times as great. The output strength is unaffected by the many divisions in the axon; it reaches each terminal button with the same intensity it had at the axon hillock. This uniformity is critical in an analogue device such as a brain where small errors can snowball, and where error correction is more difficult than in a digital system.

Each terminal button is connected to other neurons across a small gap called a synapse. The physical and neurochemical characteristic of each synapse determines the strength and polarity of the new input signal. This is where the brain is the most flexible, and the most vulnerable. Changing the constitution of various neuron- transmitter chemicals can increase or decrease the amount of stimulation that the firing axon imparts on the neighboring dendrite. Altering the neurotransmitters can also change whether the stimulation is excitatory or inhibitory. Many drugs such as alcohol and LSD have dramatic effects on the production or destruction of these critical chemicals. The infamous nerve gas sarin can kill because it neutralizes a chemical (acetyl cholinesterase) that is normally responsible for the destruction of a neurotransmitter (acetylcholine). This means that once a neuron fires, it keeps on triggering all the neurons in the vicinity. One no longer has control over muscles, and suffocation ensues, each of these neurons can connect with up to 200,000 other neurons, although 1,000 to 10,000 are typical.

1.11 From Human Neurons to Artificial Neurons

We conduct these neural networks by first trying to deduce the essential features of neurons and their interconnections as in figure 1.5.We then typically program a computer to simulate these features. However because our knowledge of neurons is incomplete and our computing power is limited, our models are necessarily gross idealizations of real networks of neurons.

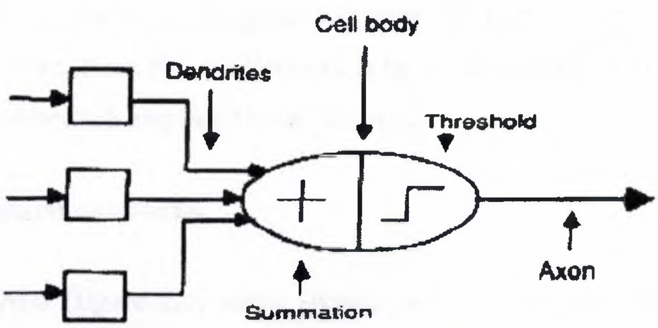

Figure 1.4. The neuron model

1.12 Summary

This chapter presented definitions of artificial neural networks, and provided brief history of neural networks since 1943 until today. And after that I have moved to give a hint about how artificial neurons they work? And I have described a different analogy between biological (human) and artificial neural networks.

CHAPTER TWO

NEURAL NETWORKS STRUCTURE

2.1 Overview

This chapter has a number of objectives. First I want to introduce the architecture of neural networks and, I define what is the meaning of learning rule, explain the perceptron network and its learning rule, and tell you how to initialize and simulate perceptron networks, also covers supervised learning and unsupervised learning networks, and provide designing a neural network, and given a application of neural networks.

2.2 Architecture of neural networks

In this section which is the Architecture Neural Network, am going to give a hint about how Neural Network 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.

2.2.1 Feed-forward networks

Feed-forward ANNs (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 ANNs 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.

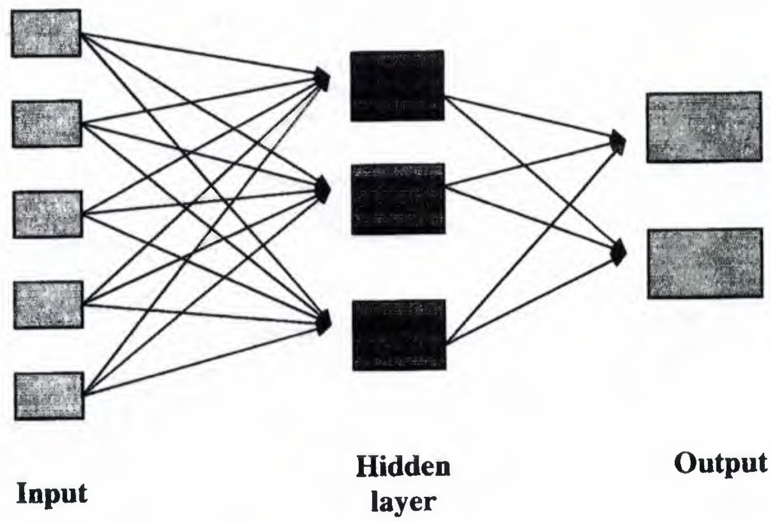

Figure 2.1. An example of a simple feed forward network

2.2.2 Feedback networks

Feedback networks (figure 2.2) can have signals traveling 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 organizations.

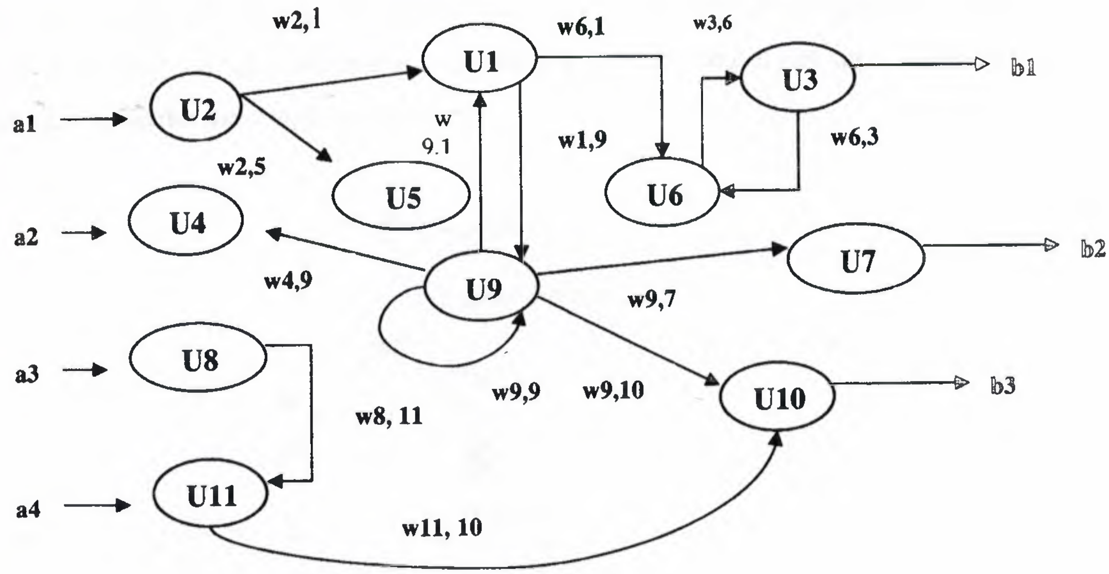

Figure 2.2. An example of a complicated network

2.3 perceptron

In the early days of artificial intelligence research, Frank Rosenblatt devised a machine called the perceptron that operated much in the same way as the human mind. Albeit, it did not have what could be called a "mental capacity", it could "learn" - and that was the breakthrough needed to pioneer today's current neural network technologies.

For a long time since perceptrons were discovered, neural networks were looked up to as powerful machines that can be trained to perform various tasks. The proof of convergence, that the perceptron will not only find suitable weights but that it will find it in finite time, was very appealing.

A perceptron is a connected network that simulates an associative memory. The most basic perceptron is composed of an input layer and output layer of nodes, each of which are fully connected to the other. Assigned to each connection is a weight which can be adjusted so that, given a set of inputs to the network, the associated connections will produce a desired output. The adjusting of weights to produce a particular output is called the "training" of the network which is the mechanism that allows the network to learn. Perceptrons are among the earliest and most basic models of artificial neural networks, yet they are at work in many of today's complex neural net applications see figure (2.3) describe the simple perceptron.

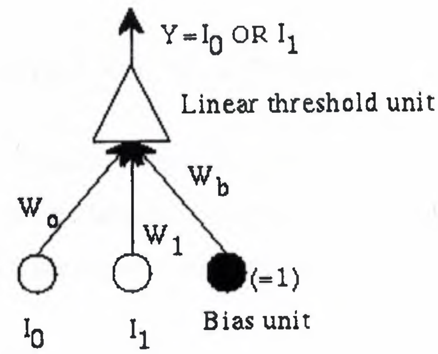

Figure 2.3. The simple perceptron

2.3.1 History of perceptron

Frank Rosenblatt invented the perceptron in 1957 at the Cornell Aeronautical Laboratory in an attempt to understand human memory, learning, and cognitive processes. On 23 June 1960, he demonstrated the Mark I Perceptron, the first machine that could "learn" to recognize and identify optical patterns.

Rosenblatt's work was a progression from the biological neural studies of noted neural researchers such as D.O. Hebb and the works of Warren McCulloch and Walter Pitts. McCulloch and Pitts had been the first to describe the concept of neural networks. They developed the MP neuron, which was based on the point that a nerve will fire an impulse only if its threshold value is exceeded. This model was somewhat of a scanning device which read pre-defined input- output associations to determine its final output. MP neurons had fixed thresholds and did not allow for learning. They were "hard-wired logic devices, which proved that networks of simple neuron-like elements could compute "[5]

Since the MP neuron did not have the mechanisms for learning, it was extremely limited in modeling the functions of the more flexible and adaptive human nervous system. D.O. Hebb suggested that "when an axon of cell A is near enough to excite cell B and repeatedly, or persistently, takes part in firing it, some growth process or metabolic change takes place in one or both cells, such that A is efficiency as one of the cells firing B is increased" [6]. This implied a "learning" network model where not only could the network make associations, but it could also tailor its responses by adjusting the weight on its connections between neurons.

2.4 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

2.4.1 Supervised Learning

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

First a pair from the training data set is chosen randomly. The input pattern of the pair is given to the network at the input layer by assigning each signal of the pattern to one neuron on this layer. Then, the network passes these signals forward to the neurons on the next layer (hidden layer). But, how is this done? For each neuron on the hidden layer, a Net Input value is computed, by doing the sum over the products of the output of each neuron on the input layer (which is the original signal itself) by the weight of the connection that connects it to the neuron on the hidden layer.

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

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

Then, these outputs are passed forward to the next layer and the same processes of computing net inputs and activations are done, until the output layer of the neural network is reached. The output values of the neurons on the output layer are taken as one pattern of signals which is considered as the actual output pattern of the network. The actual output pattern that the network produces for each input pattern is compared to the target output pattern it should have produced which is simply the second element of the example pair chosen randomly at the beginning of the whole process. An error value is computed using the actual and target patterns as follows:

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

Where

E_p: is the error value that corresponds to example pair p

O_{pi}: is the output value of neuron on the output layer of the network

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

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

Supervised learning divided into two parts:

1-Feedback nets:

A-Back propagation through time

B-Real time recurrent learning

C-Recurrent extended kalman filter

2-Feed forward –only net:

A-Perceptron

B-Adeline, Madeline

C-Time delay neural network

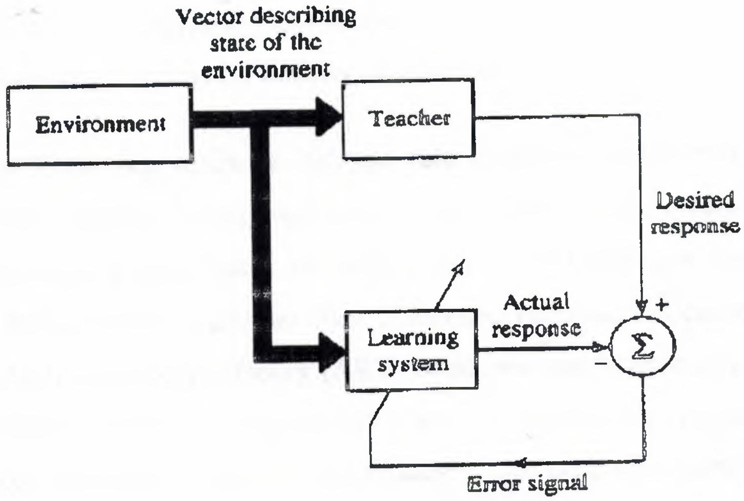

Figure 2.4. Supervised learning

2.4.2 Unsupervised Learning

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

In unsupervised or self-organized learning, the network is not given any external Indication as to what the correct responses should be nor whether the generated responses are right or wrong. It is simply exposed to the various input-output pairs and it learns by the environment, that is, by detecting regularities in the structure of input patterns.

So, unsupervised learning aims at finding a certain kind of regularity in the data represented by the exemplars. Roughly speaking, regularity means that much less data are actually required to approximately describe or classify the exemplars than the amount of data in exemplars. Examples exploiting data regularity include vector quantization for data compression and Karhunen-Loeve expansion (often referred to as principal component analysis) for dimension reduction.

In unsupervised learning, a simple Hebbian rule (correlation rule) may be applied to calculate weight changes. Energy-minimizing networks provide a recent example of unsupervised learning that makes interesting use of a two-phrase learning method. Competitive learning rules is another class of learning rules used in unsupervised neural networks. Adaptive resonance theory (ART) combines competitive and Hebbian rules together and uses feedback from the output layer to the input layer to ensure a consistent Categorization. In an ART system, connections run in both directions, from input to output nodes and vice versa. Competitive learning is used to change weights on connections from the input to the output layer in creating groupings of the input patterns.

Hebbian pattern-association learning is used to change weights on connections from the output to the input layer. As a result, an input pattern evokes a pattern on the output layer, which in turn projects the prototype of the winning group back into the input layer.

Every N.Ns goes through three operative phases:

1- Learning (training) phase – network learns on the training sample, the weights are being adjusted in order to minimize the objective function (for example RMS - root mean Square error).

2- Testing phase – network is tested on the testing sample while the weights are fixed.
3- Operative (recall) phase – NN is applied to the new cases with unknown results. When an axon of cell A is near enough to a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic changes take place in one or both cells such that A's efficiency as one of the cells firing B is increased.

Hebb proposed this change as a basis of associative learning (at the cellular level), which would result in an enduring modification in the activity pattern of a spatially distributed "assemble of never cells."

Unsupervised divided into two parts:

1-Feedback nets:

A-Discrete hop filed

B-Analog adaptive resonance theory

C-Additive gross berg

2-Feed forward –only nets

A-Learning matrix

B-Linear associative memory

C-Counter propagation

Clustering data:

Exactly one of a small number of output units comes on in response to an input.

Reducing the dimensionality of data:

Data with high dimension (a large number of input units) is compressed into a lower dimension (small number of output units).

Although learning in these nets can be slow, running the trained net is very fast - even on a computer simulation of a neural net.

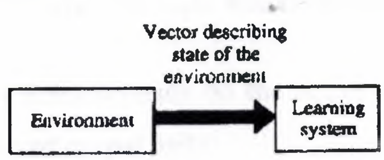

Figure 2.5. Unsupervised learning

Unsupervised divided into two parts:

1-Feedback nets:

A-Discrete hop filed

B-Analog adaptive resonance theory

C-Additive gross berg

2-Feed forward –only nets

A-Learning matrix

B-Linear associative memory

C-Counter propagation

Clustering data:

Exactly one of a small number of output units comes on in response to an input.

Reducing the dimensionality of data:

Data with high dimension (a large number of input units) is compressed into a lower dimension (small number of output units).

Although learning in these nets can be slow, running the trained net is very fast - even on a computer simulation of a neural net.

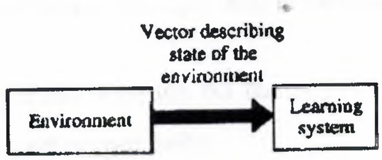

Figure 2.5. Unsupervised learning

2.5 Design a Neural Network

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

Designing a neural network consists of:

1. Arranging neurons in various layers.

2. Deciding the type of connections among neurons for different layers, as well as among the neurons within a layer.

3. Deciding the way a neuron receives input and produces output.

4. Determining the strength of connection within the network by allowing the networks learn the appropriate values of connection weights by using a training data set.

2.5.1 Layers

The commonest type of artificial neural network consists of three groups, or layers, of units: a layer of "input" units is connected to a layer of "hidden" units, which is connected to a layer of "output" units. As in Figure 2.1

The activity of the input units represents the raw information that is fed into the network.

The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units.

The behavior of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

This simple type of network is interesting because the hidden units are free to construct their own representations of the input. The weights between the input and hidden units determine when each hidden unit is active, and so by modifying these weights, a hidden unit can choose what it represents.

We also distinguish single-layer and multi-layer architectures. The single-layer organization, in which all units are connected to one another, constitutes the most general case and is of more potential computational power than hierarchically structured multi-layer organizations. In multi-layer networks, units are often numbered by layer, instead of following a global numbering.

2.5.2 Communication and Types of Connections

Neurons are connected via a network of paths carrying the output of one neuron as input to another neuron. These paths is normally unidirectional, there might however be a two-way connection between two neurons, because there may be another path in reverse direction. Neuron receives input from many neurons, but produces a single output, which is communicated to other neurons. The neuron in a layer may communicate with each other, or they may not have any connections. The neurons of one layer are always connected to the neurons of at least another layer.

2.5.2.1 Inter-Layer Connections

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

Fully connected: Each neuron on the first layer is connected to every neuron on the second layer.

Partially connected: A neuron of the first layer does not have to be connected to all neurons on the second layer.

Feed forward: The neurons on the first layer send their output to the neurons on the second layer, but they do not receive any input back form the neurons on the second layer.

Bi-directional: There is another set of connections carrying the output of the neurons of the second layer into the neurons of the first layer.

Hierarchical: if a neural network has a hierarchical structure, the neurons of a lower layer may only communicate with neurons on the next level of layer.

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

2.5.2.2 Intra-Layer Connections

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

Recurrent: the neurons within a layer are fully- or partially connected to one another. After these neurons receive input form another layer, they communicate their outputs with one another a number of times before they are allowed to send their outputs to another layer. Generally some conditions among the neurons of the layer should be achieved before they communicate their outputs to another layer.

On-center/off surround: A neuron within a layer has excitatory connections to itself and its immediate neighbors, and has inhibitory connections to other neurons. One can imagine this type of connection as a competitive gang of neurons. Each gang excites it self and its gang members and inhibits all members of other gangs. After a few rounds of signal interchange, the neurons with an active output value will win, and is allowed to update its and its gang member's weights. (There are two types of connections between two neurons, excitatory or inhibitory. In the excitatory connection, the output of one neuron increases the action potential of the neuron to which it is connected. When the connection type between two neurons is inhibitory, then the output of the neuron sending a message would reduce the activity or action potential of the receiving neuron. One causes the summing mechanism of the next neuron to add while the other causes it to subtract. One excites while the other inhibits.)

2.5.3 Learning

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

The training method usually consists of one of three schemes:

1. Unsupervised learning

The hidden neurons must find a way to organize themselves without help from the outside. In this approach, no sample outputs are provided to the network against which it can measure its predictive performance for a given vector of inputs. This is learning by doing.

2. Reinforcement learning

This method works on reinforcement from the outside. The connections among the neurons in the hidden layer are randomly arranged, then reshuffled as the network is told how close it is to solving the problem. Reinforcement learning is also called supervised learning, because it requires a teacher. The teacher may be a training set of data or an observer who grades the performance of the network results. Both unsupervised and reinforcement suffers from relative slowness and inefficiency relying on a random shuffling to find the proper connection weights.

3. Back propagation

This method is proven highly successful in training of multilayered neural nets. The network is not just given reinforcement for how it is doing on a task. Information about errors is also filtered back through the system and is used to adjust the connections between the layers, thus improving performance.

2.5.3.1 Off-line or On-line

One can categorize the learning methods into yet another group, off-line or on-line. When the system uses input data to change its weights to learn the domain knowledge, the system could be in training mode or learning mode. When the system is being used as a decision aid to make recommendations, it is in the operation mode, this is also sometimes called recall.

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

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

2.5.3.2 Learning Laws

There are a variety of learning laws which are in common use. These laws are mathematical algorithms used to update the connection weights. Most of these laws are some sort of variation of the best known and oldest learning law, Hebb's Rule. Man's understanding of how neural processing actually works is very limited. Learning is certainly more complex than the simplification represented by the learning laws currently developed. Research into different learning functions continues as new ideas routinely show up in trade publications etc. A few of the major laws are given as an example below.

Hebb's Rule: The first and the best known learning rule were introduced by Donald Hebb. The description appeared in his book the organization of Behavior in 1949[7]. This basic rule is: If a neuron receives an input from another neuron and if both are highly active (mathematically have the same sign), the weight between the neurons should be strengthened.

Hopfield Law: This law is similar to Hebb's Rule with the exception that it specifies the magnitude of the strengthening or weakening. It states, "If the desired output and the input are either active or both inactive, increment the connection weight by the learning rate, otherwise decrement the weight by the learning rate." (Most learning functions have some provision for a learning rate, or learning constant. Usually this term is positive and between zero and one.)

The Delta Rule: the data rule is a further variation of Hebb's Rule, and it is one of the most commonly used. This rule is based on the idea of continuously modifying the strengths of the input connections to reduce the difference (the delta) between the desired output value and the actual output of a neuron. This rule changes the connection weights in the way that minimizes the mean squared error of the network. The error is back propagated into previous layers one layer at a time. The process of back-propagating the network errors continues until the first layer is reached. The network type called Feed forward, Back-propagation derives its name from this method of computing the error term. This rule is also referred to as the Windrow-Hoff Learning Rule and the Least Mean Square Learning Rule.

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

The Kohonen rule does not require desired output. Therefore it is implemented in the unsupervised methods of learning. Kohonen has used this rule combined with the on-center/off-surround intra- layer connection, to create the self-organizing neural network, which has an unsupervised learning method.

2.6 The Back Propagation Algorithm

The best-known example of a neural network training algorithm is back propagation. Modern second-order algorithms such as conjugate gradient descent and Levenberg-Marquardt are substantially faster (e.g., an order of magnitude faster) for many problems, but back propagation still has advantages in some circumstances, and is the easiest algorithm to understand. We will introduce this now, and discuss the more advanced algorithms later. There are also heuristic modifications of back propagation which work well for some problem domains, such as quick propagation and Delta-Bar-Delta and are also included in ST Neural Networks.

In back propagation, the gradient vector of the error surface is calculated. This vector points along the line of steepest descent from the current point, so we know that if we move along it a "short" distance, we will decrease the error. A sequence of such moves (slowing as we near the bottom) will eventually find a minimum of some sort. The difficult part is to decide how large the steps should be.

Large steps may converge more quickly, but may also overstep the solution or (if the error surface is very eccentric) go off in the wrong direction. A classic example of this in neural network training is where the algorithm progresses very slowly along a steep, narrow, valley, bouncing from one side across to the other. In contrast, very small steps may go in the correct direction, but they also require a large number of iterations. In practice, the step size is proportional to the slope (so that the algorithm settles down in a minimum) and to a special constant: the learning rate. The correct setting for the learning rate is application-dependent, and is typically chosen by experiment; it may also be time-varying, getting smaller as the algorithm progresses.

The algorithm is also usually modified by inclusion of a momentum term: this encourages movement in a fixed direction, so that if several steps are taken in the same direction, the algorithm "picks up speed"[8], which gives it the ability to (sometimes) escape local minimum, and also to move rapidly over flat spots and plateaus. The algorithm therefore progresses iteratively, through a number of epochs. On each epoch, the training cases are each submitted in turn to the network, and target and actual outputs compared and the error calculated. This error, together with the error surface gradient, is used to adjust the weights, and then the process repeats. The initial network configuration is random and training stops when a given number of epochs elapse, or when the error reaches an acceptable level, or when the error stops improving

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) [9], 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 NN (that is, input data without the desired result), and the NN 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 feed-forward 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, feed-forward networks with a single hidden layer and trained by least-squares are statistically consistent estimators of arbitrary square-inerrable 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 .Feed-forward 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 large 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 N.Ns that can magically create information that is not contained in the training data.

2.8 Where are Neural Networks Being Used?

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

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

This is particularly useful with sensory data or with data form a complex (e.g. chemical, manufacturing, or commercial) process. They may be an algorithm, but it is not known, or has too many variables .it is easier to let the network learn from examples

Neural networks are being used:

1. In investment analysis: to attempt to predict the movement of stocks currencies etc., from previous data. There, they are replacing earlier simpler linear models.

2. In signature analysis: as a mechanism for comparing signatures made (e.g. in a bank) with those stored. This is one of the first large-scale applications of neural networks in the USA, and is also one of the first to use a neural network chip.

3. In process control: there are clearly applications to be made here: most processes cannot be determined as computable algorithms. Newcastle University Chemical Engineering Department is working with industrial partners (such as Zeneca and BP) in this area.

4. In monitoring: networks have been used to monitor the state of aircraft engines. By monitoring vibration levels and sound, early warning of engine problems can be given. .British Rail has also been testing a similar application monitoring diesel engines.

5. In marketing: networks have been used to improve marketing mailshots. One technique is to run a test mailshot, and look at the pattern of returns from this. The idea is to find a predictive mapping from the data known about the clients to how they have responded. This mapping is then used to direct further mailshots

2.9 Summary

In this chapter I tried to concentrate on the algorithms and the structure of the neural network, here there are two categories, which are Feed-forward Network and Feedback Network. And after that I have to moved ahead to give a hint about how the neural network can be trained. The neural network able to be trained in two methods the first one is Supervised Learning and the second one is Unsupervised Learning.

CHAPTER THREE

NEURAL NETWORK IN MEDICINE

3.1 Overview

The aim of this chapter is to demonstrate some basic applications of neural networks in medicine. It presents understanding the concept of a diagnosis prediction via an artificial neural network knowledge base. One of the major applications of medical information has been the metabolic disease diagnosis, and the general theme of this chapter is the ethical issues in the use of neural network, based methodologies for image interpretation in medicine, I will discuss a clinical application also, finally the chapter ends with artificial neural network in immunology.

3.2 Integration of Neural Networks and Knowledge-Based Systems in Medicine

Knowledge-Based Systems are used in medical diagnoses. They have the advantage to give an explanation of a diagnosis. But a main problem when dealing with Knowledge-Based Systems is the acquisition of knowledge. Artificial Neural Networks deal with knowledge in a sub symbolic form. Incomplete and imprecise data can be processed by approximating not linear relations in data. In a laboratory or medical system the integration of the neural network system into the decision making process may be required. We realized this by building a hybrid system consisting, first, of graphical visualizations methods and second, a machine learning module generating rules out of the neural network. The rules are presented in a form, which can be understood by humans and used in Knowledge-Based Systems. Keywords: Knowledge-Based System, Neural Network, decision making, visualization, machine learning. Integration of Neural Networks and Knowledge-Based Systems

Knowledge-Based Systems are used in medical diagnoses. They have the advantage to give an explanation of a diagnosis. This is very important especially in the domain of medicine where the user wants to have the diagnosis proved. But a main difficulty when dealing with Knowledge-Based Systems is the acquisition of the domain knowledge. There are several problems with it. It is difficult to transform the explicit and implicit knowledge of the expert's domain, which also partly consists of own experience, in a form which is suitable for a knowledge base. The knowledge can also be inconsistent or incomplete. A second problem is that Knowledge-Based Systems are not able to learn from experience or to operate with cases not represented in the knowledge base.

Artificial Neural Networks deal with knowledge in a sub symbolic form. They can solve non-linear problems often better than conventional methods and are capable to approximate non linear relations in data. In addition, incomplete and imprecise data can be processed. Neural networks learn in a massively parallel and self-organizing way. Unsupervised learning neural networks, like Kohonen's self organizing feature maps [10], learn the structure of high-dimensional data by mapping it on low dimensional topologies, preserving the distribution and topology of the data. But large neural networks can only be interpreted with analyzing tools. We developed a visualization method, the so called U-Matrix methods, to detect the structure of large two-dimensional Kohonen maps. It generates a three-dimensional landscape on the map, whereby valleys indicate data which belongs together and walls separate subcategories [11].

In a laboratory or medical system the integration of the neural network system into the decision making process may be useful. The knowledge of neural networks, however, is in this form not communicable; i.e. it is necessary to transform the knowledge into a form, which, first, can be understood by humans and second, can be processed by knowledge based systems. Knowledge based systems have the advantage that they can give an explanation of a diagnosis. By integrating both paradigms, knowledge based systems and neural networks, the disadvantages of both approaches can be redressed. We are developing a hybrid system REGINA (rule extraction and generation in neural architecture) which consists of several parts. An unsupervised learning neural network maps the (preprocessed) data space onto a two-dimensional grid of neurons, whereby it preserves the distribution and topology of the input space. But only together with a visualization module, called U-Matrix methods, we are able to detect structure in the data and classify it. A three-dimensional colored landscape will be generated in which walls separate distinct subclasses and subcategories are represented by valleys. A machine learning algorithm extracts rules out of the learned neural network [12]. In distinction to other machine learning algorithms like ID3 our algorithm considers the attributes by selecting those which are relevant for the classification. This corresponds to the proceeding of a medical expert. The rules can be used as a knowledge base for an expert system. Also fuzzy rules can be extracted out of the neural network.

3.3 MODELLING OF THE HUMAN ARTERIAL NETWORK FOR PREOPERATIVE PREDICTIONS

Most times modeling and simulation in human medicine is a very difficult task to do, because the human body is a complex system involving physical and chemical processes. Sometimes only inaccurate or even no explanations are available for things going on in the human body. The fact, that the physiology differs from person to person, makes it more difficult to develop general models which are valid for a large group of persons.

The main purpose of the model of the human arterial network is to describe the relationship of the morphology and the hydraulics. This model offers methods to calculate mean flow velocity, mean flow, flow direction and blood pressure in the arteries. The results are accurate enough to make fundamental statements on blood supply of specific parts of the arterial system and on the flow velocity through the arteries.

3.3.1 Physiology of the Human Arterial System

The vessels -- arteries, capillaries and veins -- together with the heart form the cardiovascular system. It is a transport system in which a pump (heart) transports blood through a closed system of flexible pipes. The main purpose of the system is to supply all cells in the organism with substances that are necessary for the normal function of the cells (e.g. oxygen, nutritive substances ...) and to carry off the metabolic substances. (Schmidt 1983) [13]

Picking out only the arterial system (capillaries excluded) of the systemic blood circulation, it can be described as follows: The arterial system is a network of flexible pipes. At one point blood is pumped into the system. The same amount of blood, that enters the network, leaves it at the end points.

The heart generates a pulsatile flow of blood through the arteries, but the compliance of the arteries smoothes this pulse. This fact causes a nearly steady flow in heart distant parts of the body.

The blood itself consists of blood plasma (56%) and different cells. Blood is not a fluid in the physical sense, but its characteristics are similar to that of a Newton fluid if it flows through large vessels.

3.3.2 Physical Model Pipe Network

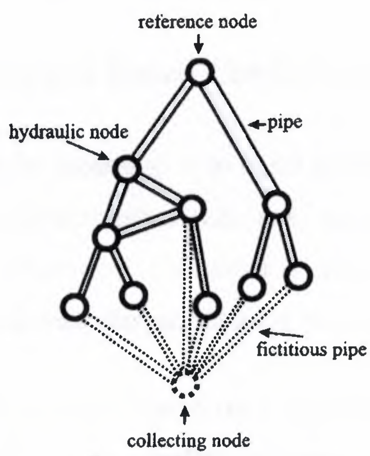

Figure 3.1. The pipe network

The physiological facts give the motivation for the identification of the arterial system with a pipe network. Every segment of an artery between two ramification points is represented by a pipe and every ramification point by a hydraulic node. The network is supplied with a constant inflow of fluid (blood) at one node. That means that the pulsatile flow from the heart and the compliance of the arteries are neglected.

The blood can leave the network at any node and flow into other parts of the arterial system or into the venous system which are not take into account. The amount of blood which is lost at the nodes is unknown. Therefore all blood is collected through fictitious arteries. (Almeder 1997) [14]

This pipe network has the following features (see figure 3.1.):

1. There is only one node with a constant flow into it and given pressure.

2. The outflow of all nodes is collected in one fictitious node. The additional node is connected to the other nodes through fictitious vessels of the same length, but with variable diameters. Those are used to regulate the resistance of other parts of the arterial system.

3. Blood is modeled as a Newtonian fluid.

4. The arteries are a hydraulic smooth pipe that means that the roughness of the walls is zero.

3.4 Artificial Neural Network Based Cardiovascular Modeling

One approach to cardiovascular modeling is to build a model representative of a group of individuals with similar characteristics (i.e., sex, age, physical condition, medical condition, etc.). However, cardiovascular behavior is unique to each individual (Vander 1990)[15], thus a generic cardiovascular model used in a medical diagnostic system

would not be as sensitive as a system based on a model that is adapted to the patient being diagnosed. To develop these models without a cardiovascular expert, the modeling must be based on an adaptive technology that can be automated. The ANN technology fits this category. The ANN technology was selected for the cardiovascular modeling because of its many capabilities including sensor fusion, which is the combining of values from several different sensors. Sensor fusion enables the ANNs 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, ANNs are capable of detecting complex medical conditions by fusing the data from the individual biomedical sensors. Recurrent ANNs were selected for the cardiovascular modeling application to capture the temporal information in physiological variables. These variables are time-series data from which both the absolute values and the rates of change need to be modeled. Recurrent ANNs recycle a small portion of information from time t-1 at time t.

Indirectly, decreasing portions of information from time t-2, t-3, t-4, etc. are also captured, thus enabling recurrent ANNs to model the temporal dynamics in data. Figure3.2 illustrates a prototype tool that generates an ANN model of the cardiovascular system from physiological variables received from biomedical sensors attached to an individual.

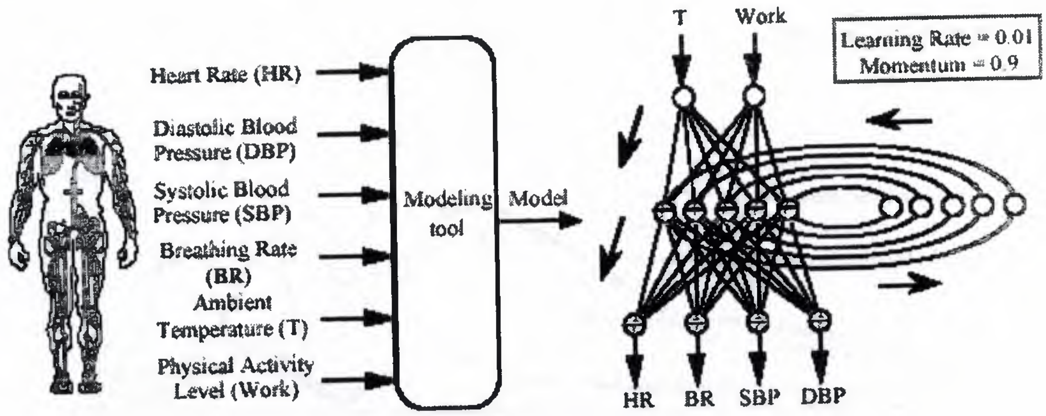

Figure 3.2. Illustrates a prototype tool that generates an ANN model

On the left, this figure (figure3.2) illustrates a modeling tool that takes a sequence of physiological variables from biomedical sensors and learns the temporal dynamics of these variables to produce an ANN-based cardiovascular model. On the right, this figure illustrates the configuration of the ANN produced by the modeling tool. The ANN has two inputs, four outputs, and five hidden processing elements. The ANN takes the ambient temperature and the physical activity as input. The four outputs, heart rate, breathing rate, systolic blood pressure, and diastolic blood pressure, are clamped to the "actual" values during the training phase. For the initial cardiovascular model prototypes, the "actual" values are generated by anon adaptive cardiovascular model. During the modeling phase, the temperature and the work are input to the ANN, and the Values at the outputs are taken as the modeled variables. The feedback links going through the five processing elements on the right side of the ANN enable it to capture temporal information in the data. During the adaptation phase, the training algorithm receives physiological data from an individual via biomedical sensors and automatically develops the ANN-based cardiovascular model. After development, the model can generate the appropriate physiological responses for simulations with varying levels of physical activity. Figure3.2 shows how the variables modeled with the ANN compare with the physiological variables generated with anon adaptive cardiovascular model. This second model has been used for creating data with sufficient complexity for the development of the modeling tool.

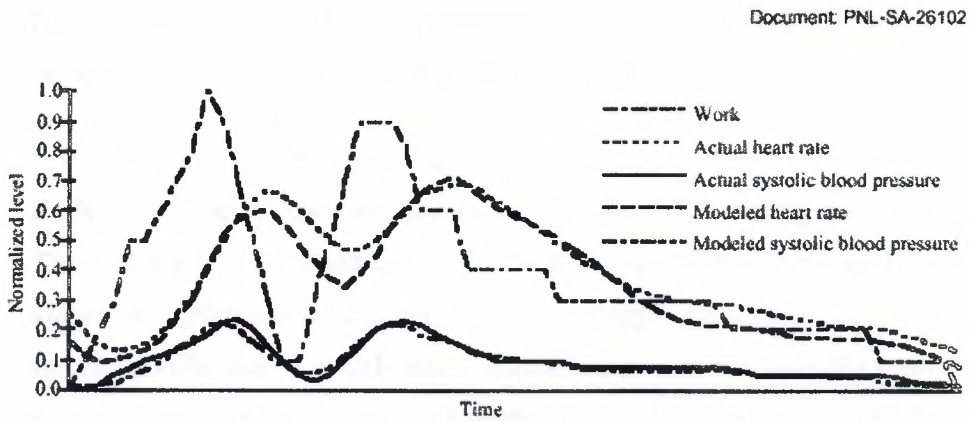

Figure 3.3.This figure depicts the "actual" and modeled heart rate, and the "actual" and modeled systolic blood pressure for varying physical activity levels.

The "actual" variables in this figure (figue3.3) are generated with a no adaptive cardiovascular model. The vertical axis corresponds to the normalized magnitude of these variables (normalized to one). The variables for systolic blood pressure and breathing rate are excluded from this figure for clarity. The effect of varying ambient Temperature has not yet been explored in this research.

3.5 Model Based Cardiovascular Diagnostics

It is envisioned that cardiovascular models will be incorporated in both clinical diagnostic systems for graded Exercise tests and cardiovascular stress tests, and in an automatic, continuous diagnostic system carried on a person. The methodology for using models as a basis for diagnosis is often referred to as "model-based reasoning."

Diagnostic systems that use model-based reasoning compare actual data to modeled data and exploit the differences for diagnosis. Two prerequisites for this methodology to be successful are that the models are authentic to the systems being diagnosed and that the differences between the modeled data and the actual data are known for diagnostic conditions. Conventional modeling techniques tend to build generic models with possibly a few free variables that fit the model to an instance of a system. For example, a respiratory system model-based on differential equations may have a few free variables adjusted to an individual's sex, age, and weight (Tehrani 1993)[16]. An ANN-based model is potentially a superior model because almost all of its free variables are adjustable to behave as a specific instance of a system.

Conventional diagnostic techniques most often require that the differences between the modeled and actual data are known to the person developing the diagnostic system. These techniques are handicapped by both the ability of the person to understand the diagnostic differences in the data and by the applicability of those differences to the modeling technique. An ANN-based diagnostic system is potentially superior because it does not require a prior knowledge of the diagnostic differences in the data, although it should be recognized that some knowledge aids the development reasoning to produce a diagnosis of health by comparing a model of an individual to the individual's current condition. A diagnostic system based on a model uses an individual's normal-condition cardiovascular behavior as reference. Any variation from that behavior indicates a change from the normal condition. An ANN-based diagnostic system is trained to recognize the effects of certain medical and physical changes on the monitored variables. For example, a blood loss results in a decrease in blood pressure and an increase in heart rate relative to the normal values for that individual. Figure 3.4 illustrates a diagnostic system and the information flow in model-based reasoning. The modeling tool receives the physiological variables from an individual via biomedical sensors.

The diagnostic system receives the same variables from both the biosensors and the model. These two sets of variables are "compared" for diagnosis.

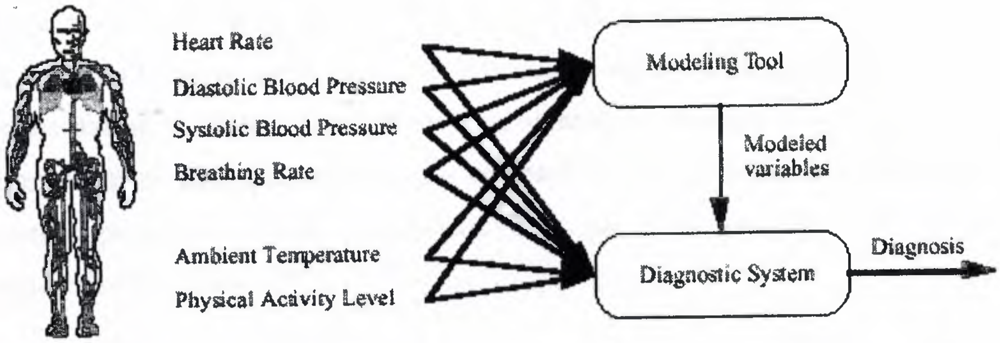

Figure 3.4. Illustrates the information flow within a cardiovascular diagnostic system that uses model-based

3.6 Clinical Application

Patients with persistent, intractable pain, exemplified by low-back and lower-extremity pain after multiple surgeries are managed in several ways. Anatomical procedures may be used to correct an abnormality thought to cause the pain. Ablative procedures may be performed to destroy neural pathways that transmit pain signals. Augmentative procedures that superimpose electrical stimulation or medications that act on the intact nervous system may also be administered. Augmentative procedures, which are reversible and amenable to straightforward trial before devices are permanently implanted, are used increasingly because of their obvious advantages.

Spinal cord stimulation is a neuroaugmentative procedure: electrodes are implanted over the dorsal surface of the spinal cord and electrical pulses are delivered from an implantable generator. At stimulus amplitudes adequate to relieve pain, this produces a parenthesis. Overlap of the topography of a patient's pain by parenthesis is a necessary condition for pain relief, and there is a highly significant association between overlap and patient ratings of pain relief. The position of the implanted electrode(s) largely determines the location of stimulation parenthesis. To maximize the likelihood of achieving the desired overlap and to minimize the need for repeated surgical revisions of electrode position, manufacturers have developed arrays of multiple contacts.

As the number of contacts has increased, however, the task of adjustment (thoroughly testing the available anode and cathode assignments) has grown enormously. For a simple contact pair, there are only two assignments to test; for four contacts, there are 50; for eight contacts, there are 6050. The time required of the patient, the physician, and the staff to test these combinations is substantial. If adjustments are made manually, representing the data quantitatively for comparisons and retesting is cumbersome and becomes prohibitively time consuming. This has motivated the development of computerized methods [17].

Even using computerized methods that require minimal patient supervision after initial training, the task of adjustment can become prohibitively time consuming if all possible combinations are to be tested exhaustively. For an eight-electrode array, even at a rate of 50 combinations per hour, it would take 121 hours to test all the possibilities. The performance of different electrode combinations may be assumed to be interdependent, and thus representative subsets may be defined and search strategies developed so that not all combinations need to be tested. Validating this assumption may be difficult for large numbers of contacts and for complex (two-dimensional) geometries. In the present study, the available 50 combinations for an array of four electrodes is a manageable number; they have been tested exhaustively, allowing validation of our models.

To date linear discriminate methods have yielded the finding that a particular electrode configuration ("guarded cathode" or "split anode," that is, cathode[s] flanked by anode[s] above and below) is preferred by our patients to a statistically significant degree.[8] In this prior work, as in the present study, the importance of overlap of pain by stimulation parenthesis in achieving the clinical goal of pain relief is assumed; this has been established before and is beyond the scope of the present study [18] .

3.7 Ethical Issues in the Use of Neural Network-based Methodologies for Image Interpretation in Medicine.

The concern about the ethical implications of the use of artificial intelligence techniques in medicine is ongoing. On the one hand, the use of artificial intelligence increasingly provides opportunities to facilitate and enhance the work of medical experts and ultimately to improve the efficiency and quality of medical care. On the other hand, the debate about the appropriate level and the role of intelligent decision support has become more complex, as technical, organizational and social issues become intertwined. In this paper we use a research project that applies neural network-based methodologies as an opportunity to study the ethical issues that may rise from the application of artificial intelligence techniques in a medical context.

Advances in neuron computing have opened the way for the establishment of decision support systems which are able to learn complex associations by example. It is acknowledged that the appropriate use of the neural network-based methodologies in medical problem solving could be very effective to improve the efficiency and the quality of medical care. The growing number of projects that employ neural network based methodologies in medical care makes necessary to examine the level and the role that neural networks will play in the development of automatic diagnostic systems.

This section considers the use of neural networks in image processing and knowledge representation in image interpretation systems from an ethical perspective. The diagnostic attributes of such systems are based on the result of the feature extraction procedure from digital images, in conjunction with previously available medical knowledge, in order to expand this knowledge. A key issue in the use of neural network methods in a medical application of this kind is that it is unclear how decisions are reached. Most neural networks suffer from the opaqueness of their learned associations. In medical applications, this black box nature may make clinicians reluctant to utilize a neural network application, no matter how great the claims made for its performance. Thus, there is need to enhance neural networks with rule extraction capabilities. In addition, it is necessary to examine how productivity can be increased and how quality can be assured. This examination addresses the specification of the problem, the development of appropriate representations for the network input and output information and the preparation of the training, testing and validation data.

Using the acquisition, processing, storage, dissemination and use stages of an information life cycle as a basis for the different stages for the development and use of neural network medical imaging applications, this paper elaborates on the broader ethical implications of the use of neural applications in medicine. These include issues of interpretation, coordination between the technology and the human expert, validation of results and professional responsibilities. Although we concentrate on the use of neural network-based methodologies, the ethical issues discussed in this paper are relevant to a broader spectrum of artificial intelligence and information technology applications in health care. To illustrate this broader set of topics, the paper concludes with suggestions for future research in the area of medical informatics that can support ethical practice.

3.8 A Neural Network Model for Metabolic Disease Diagnosis

We have developed a prototype computer program, MetaNet, that uses combination of artificial neural networks and knowledge-based expert systems to assis 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 knowledge-based expert system also includes a section entitled Independent Metabolic Disease Reference Documents, which provides additional information about a suspected metabolic disease when requested by the user. 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.

3.9 Neural Network for Plasma Torch Operation and Process Control

The prediction and classification features of ANN have been used to support torch control at DIAL, and other potential applications are being identified. ANN currently has a modular design to use it as a component in a larger intelligent support system.

ANN derives its topology from the interconnections of neurons in the human brain. It learns the characteristics of each process by observation, and adapts its strategy for a particular situation. Neural networks automatically take advantage of process upgrades and compensate for sys-tem degradation. Nonlinear and multivariate capabilities of neural networks make it ideal for even direct control. The network's neurons (represented by the filled circles) work together to make decisions based on their individual inputs and weights. The network is trained by sampling thousands of examples of actual data where input patterns were received and correct decisions were made the network learns to recognize these correct patterns using a neural mathematical model. After being trained to recognize input patterns, the neural network can make good decisions even for new and unfamiliar patterns.

Neural networks handle time series prediction in a unique way as they are versatile and accommodate nonlinear data with noise. It adjusts its own model based on the behavior of historical data, to predict future behavior fairly accurately. As many of the time series have a significant chaotic component, neural networks do a far better job in handling them than other models of time series prediction.

Time series prediction techniques have been applied at DIAL to plasma torch voltages to stabilize output power using the back propagation model of an artificial neural net-work. The torch power fluctuation is caused by the variation of gas pressure inside the electrode to move the arc attachment point. Operation with small power fluctuations is not usually detrimental. However, power stabilization allows more freedom for gas pressure variation, and this can be very useful. The application of ANN to solve this problem has eliminated the power variation due to gas pressure for the trained operating conditions. During classification, a fully trained ANN recognizes a familiar pattern or makes an approximate guess on the unfamiliar patterns. DIAL has also applied ANN to characterize a simulated waste feed stream. Physical properties of waste materials have been used as inputs to ANN to identify the materials. In the present model, a fuzzy

ARTMAP network uses four parameters to identify ten materials. The system classified 94% of the test data correctly.

3.10 Application of Neural Network in Immunology

The immune system recognizes agents foreign to the host organism and raises appropriate responses. Foreign includes viruses, bacteria, parasites, fungi, tumors, and transplants. The immune recognition process in vertebrates involves major histocompatibility complex (MHC) molecules which bind short peptides and display them on the cell surface for recognition by T-cells of the immune system [19]. Binding of peptides to MHC molecules is necessary for immune recognition, but only a limited number of peptides can bind to a specific MHC molecule. Determining which peptides can bind to specific MHC molecules is crucial for understanding the basis of immunity and is important for identifying of candidates for the design of vaccines and immunotherapeutic drugs.

Prediction of peptide binding to MHC molecules is a combinatorial pattern recognition problem. A peptide is a chain composed of amino acids. The majority of the peptide cores that are responsible for binding to MHC molecules are 9 amino acids long, although lengths of 8- 12 amino acids are also common. There are 5.12´10 11 peptides of length nine that can be composed of twenty naturally occurring amino acids. Peptide sets specific for a particular MHC variant in some cases overlap sets specific for other MHC variants. Often, these sets are exclusive. Human MHC is known as human leukocyte antigen (HLA); more than 700 variants have been characterized to date. An individual may express up to 20 different HLA molecules. An immune system has to discriminate between self and foreign peptides, tolerate more than 10 7 self peptides and respond appropriately to a relatively small subset of targets from more than 10 11 potential foreign peptides. For a given protein the peptide targets of immune recognition often differ between individuals even when some of their HLA molecules are identical [20].

The importance of the computational analysis in immunology is increasing with recent advances in molecular and clinical immunology. These advances have resulted in accumulation of experimental data and in improved understanding of immunological processes. Immunological databases are growing in both size and complexity [21]. This complexity growth has roots in the combinatorial aspects of immunology: genes that encode products of the immune system are the most variable (polymorphic) gene super family of the organism. The immune system products interact with, and facilitate appropriate responses against, practically unlimited number of targets. The immune response targets include peptides derived from environmental-, microbial- or self-antigens. The discovery of specific targets of immune responses is important, but is becoming increasingly difficult as we learn more about the complexity of the domain. hidden Markov models (HMMs), and molecular modeling. Computational models have been used in immunology to a) minimize the number of experiments required for the determination of targets of immune responses, or b) to conduct large-scale computational simulations facilitating knowledge discovery when the experimental approach is not possible. Computational methods for data analysis and modeling in immunology must provide the ability to a) deal with fuzzy data, b) deal with incomplete data, c) tolerate noise and errors, and d) easily incorporate new data.

Models based on ANNs have played a prominent role in computational immunology applications. ANN-based models have proven superior in accuracy and ease of both application and model refinement. The aims of this article are a) to discuss the use of computational models in immunology, and b) to describe successful applications of ANN models in immunology.

3.11 Summary

This chapter presented applications of neural networks in medicine. The chapter described the integration of neural networks and knowledge based system in medicine, and presented the clinical application, then I introduce artificial neural network based cardiovascular modeling, I discuss model neural network for metabolic Disease Diagnosis, and the artificial neural network application for plasma torch operation and process control.

CHAPTER FOUR

Pulse-Coupled Neural Networks for Medical Image Analysis

4.1 Overview

This chapter that performed on initial evaluation into the application of pulse-couple neural networks (PCNNs). Since there is much research in the application of advanced image processing techniques to medical imagery, it is choose the medical imagery for this examination. Also, there are many medical image modalities and a wealth of image sources. It is also choose sets of MIRI brain and abdomen images and a set of pulmonary scintigraphic images to represent two extreme cases.

4.2 Pulse-Coupled Neural Networks

A PCNN is a physiologically motivated information-processing model based on the mammalian visual cortex. The underlying model was proposed by Eckhorn to explain the experimentally observed pulse synchrony process found in the cat visual cortex. This model is significantly different than other artificial neural network models in both its structure and operation. In the PCNN model, each neuron in the processing layer is directly tied to an image pixel or set of neighboring image pixels. Each neuron iteratively processes signals feeding from these nearby image pixels (i.e., feeding inputs) and linking from nearby neurons (i.e., linking inputs) to produce a pulse train. There is no training involved for the PCNN. Figure 4.1 illustrates the major structures in the PCNN. Similarities in the input pixels cause the associated neurons to fire in synchrony indicating similar structure or texture. This synchrony of pulses is then used to segment similar structures or textures in the image. In our study, we implemented two PCNN algorithms. The first was based on the work of Shane Abrahmson . The second was based on the work of Jason Kinser.

4.3 Segmentation From Magnetic Resonance Imagery

This experiment that performed was to determine the effectiveness of PCNNs on high quality (i.e., low noise and uniform contrast) images. It is choose sets of magnetic resonance images of the brain and of the abdomen because they are good representations of high quality medical images.

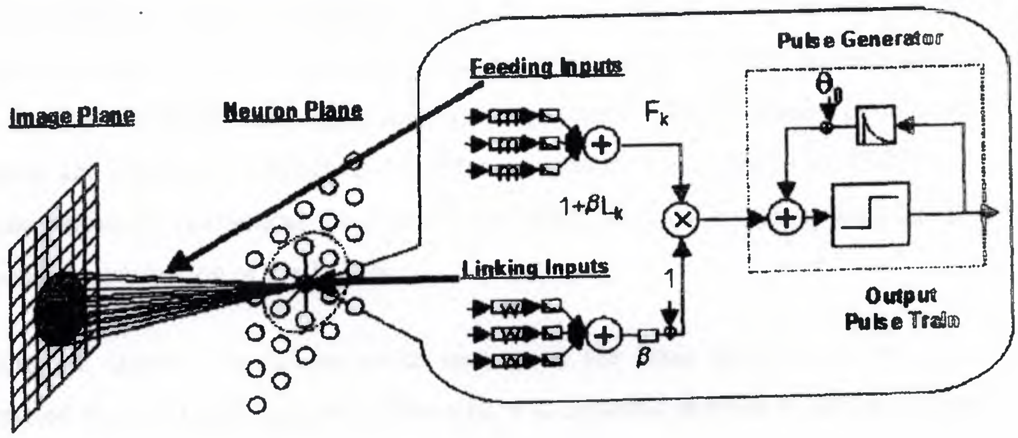

Figure 4.1. This figure illustrates the major components of the pulse-coupled neural network.

4.3.1 Magnetic Resonance Imaging

Magnetic resonance imaging (MIRT) is a popular diagnostic tool that allows the physician to examine internal anatomical structures (e.g., organs, muscles, blood vessels, and other soft tissue). It produces images of good contrast and relatively low noise. The process essentially involves determining the density of hydrogen nuclei within the body. This is done by placing the subject in a magnetic field and sweeping a radio frequency across the resonance frequency of the magnetic spins of hydrogen nuclei . MIRI is not tied specifically to hydrogen, but since it is a common element (often bonded with oxygen in water or with carbon in hydrocarbons), it produces a good representation of the internal body structure. This representation is three-dimensional, but is often visually represented as a sequence of two-dimensional slices.

MIRI has excellent soft tissue differentiation yielding a good boundary contrast between anatomical structures. It is most commonly used in the evaluation of patients suspected of having tumors, multiple sclerosis, inflammatory diseases, disc disease, cartilage and soft tissue abnormalities, and tendon problems.

4.3.2 Automated Segmentation

In MRI, tumors show up as either areas of denser tissue mass, which results in a different intensity within the image, or as an enlarged area. While manual evaluation works well to identify and locate tumors, image segmentation is useful to calculate the volume and evaluate the progression of tumor growth. We chose to try PCNNs on the segmentation of anatomical structures from MIRI because it has a potential benefit beyond the evaluation of the PCNN.

Figure 4.2 shows a horizontal cross section of the brain along with the segments extracted by the PCNN algorithm. Visually, it is apparent that the PCNN is performing an adaptive thresholding process revealing major brain structures.

Figure 4.2. This figure shows a MRT horizontal cross sectional image of the brain (far left), segments extracted (middle), and combined PCNN image highlighting major segments of the brain (far right).

4.3.3 Evaluation of PCNN Segmentation

The segmentation process underlying PCNNs uses a cross sectional view of the abdomen since it represented a more complex image than the brain. Figure 4.3 shows the MRT cross sectional image of the abdomen and the resulting image generated by the PCNN. This is an image of a person with an enlarged right kidney. While the physician can easily see that the kidney is enlarged, image segmentation is used to determine a quantitative figure for the kidney size. Figure 4.5 shows the sequence of segments generated by the PCNN process. Varying the many PCNN parameters generates a wide range of sequences.

To verify that the PCNN was doing something more than just thresholding, the relationship between the input pixels and output pixels examined. Figure 4.5 show the mapping range from the original input image to the processed output image. Is show that the range of pixel values in the input image mapping to a single output pixel value overlaps other mapping ranges. This validates the PCNNs are doing more than just level thresholding and that PCNNs are combing spatial information in the segmentation process. Figure 4.6shows histograms of the pixels in the entire image and the distribution of pixels being transformed into output pixel values. This figure also illustrates that more than pixel value is used to produce a segment.

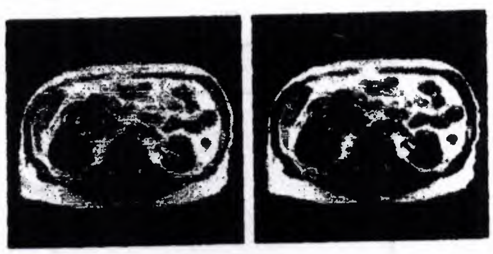

Figure 4.3. This figure shows a MIRI cross sectional image (left) of the abdominal region within an enlarged right kidney (left side of image) and the PCNN processed image (right).

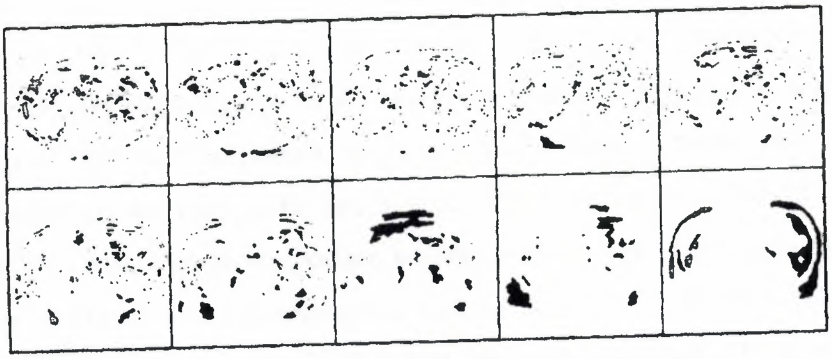

Figure 4.4. This figure shows the sequence of 10 non-black segments making up the composite output PCNN image of the abdominal region shown in Figure 3 These segments can be combined to assist in the segmentation of organs and other body tissue.

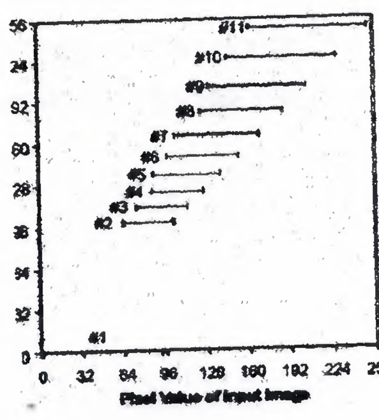

Figure 4.5. This figure shows the range of pixels values on the input image that are converted to a single pixel value on the output image. Unique pixel values in the output image represent different segments. This figure illustrates that there is great overlap of pixel ranges validating that PCNNs are an adaptive thresholding algorithm that incorporates spatial information.

Image segmentation is used to evaluate the size of anatomical structures. For the cross sectional image of the abdomen the normal left kidney with the abnormal right kidney composed. By combing the PCNN segments shown in Figure 4, a segmentation map was generated which separates the organs and abdominal cavity from the rest of the abdominal tissue including fat and muscle. Simple thresholding was also performed on the MIRJ cross section to produce a segmentation map. The segmentation maps are shown in Figure 4.6. The segmented regions representing the two kidneys were then manually separated from each other and the rest of the map. The pixel count for the kidneys in each of segmentation map was measured. Next, the kidney was manually segmented from the original cross sectional image of the abdomen. The ratio of the abnormal kidney to the normal kidney was approximately 3 +0.2. While these results might seem promising, they came with a good deal of manual processing. The PCNN parameters were manually adjusted and the kidneys segments had to be manually separated from the rest of the abdominal structures. To produce a truly automated approach, much additional work is necessary.

Figure 4.7 shows an idealized segmentation of the kidneys. This was achieved by running the PCNN with a wide linking radius (8 pixels) followed by an adaptive process of segment combination and spatial filtering with a smoothing filter and a median window filter. Again, this result came through manual intervention and much trial-and-error.

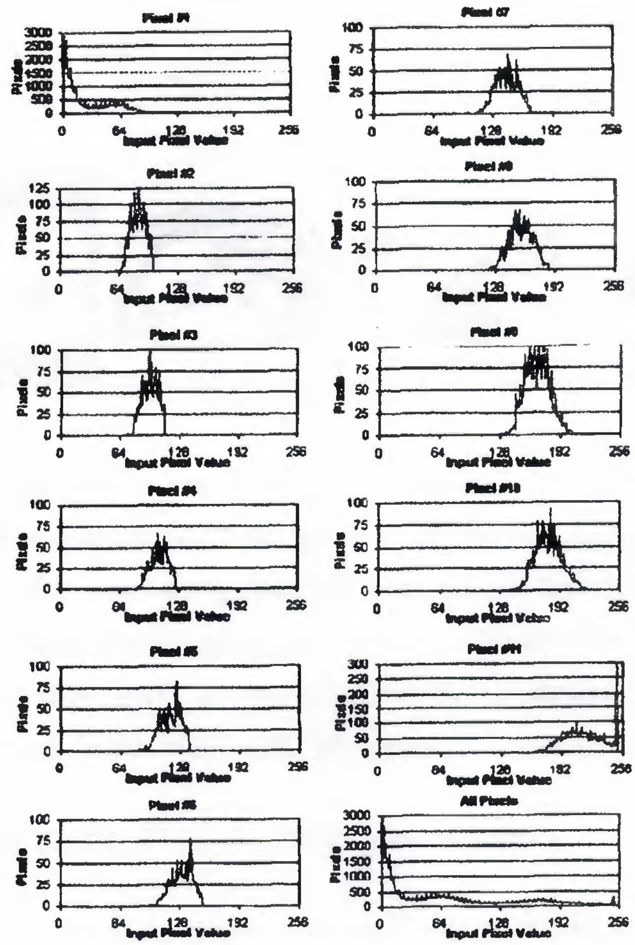

Figure 4.6. Histograms of the pixels in the entire image and the pixels being transformed into the 11-output pixel values.

Figure 4.7. This figure compares the final segmentation of the abdominal region. The image on the left represents the segmentation produced by the PCNN. The images in the middle and on the right represent straight thresholding of the raw images. White regions

indicate fat and muscle tissue. Black regions indicate organs, cavities, vertebrate, and the region outside the body.

Figure 4.8.This figure represents another processing of the abdominal region by another PCNN set with a wide linking field followed by adaptive segment combination and spatial filtering with a smoothing filter and a median window filter. On the left is the complete image highlighting three major segments in the abdomen: kidneys (dark gray), abdominal cavity (light gray), and fat and muscle tissue (white). These three segments are shown separately in the middle and right frames.

4.4 Lung Diagnostics from Scintigraphic imagery

The second set of experiments that performed was to determine the effectiveness of PCNNs on noisy images with varying contrasts. It is chosed a set of nuclear scintigraphic images of the lungs (i.e., pulmonary scintigrams) because they are a good representation of noisy images that must be used directly to make medical diagnoses. Since this diagnostic procedure requires the use of nuclear materials that emit gamma rays, the dose levels for the radioactive tracer are minimized to the lowest image detection threshold. This results in images with low signal-to- noise ratios and poor uniformity.

4.4.1 Pulmonary Scintigraphic or V/Q Imaging

Scintigraphic imaging (i.e., nuclear medicine imaging) is commonly used to evaluates body organs, functions, and abnormalities. When a patient is suspected of having a respiratory problem, he or she is subjected to a lung ventilation scan

(V) and lung perfusion scan (Q). This procedure is known as a VIQ scan or pulmonary scintigraphy. The physician looks at two things: the amount and distribution patterns of blood in the lungs (perfusion), and the amount and distribution of air throughout the lungs (ventilation). The physician identifies regions of the lungs that are not receiving the normal amounts of air, or blood, or both, as these indicate an abnormal lung condition. This involves the patient inhaling a gas that is laced with a fine radioactive dust or radioactive smoke (usually Technetium-99 or Xenon). Images of the lungs are then taken at several different angles around the long axis of the patient with a gamma camera (i.e., gamma-ray imaging system). The images are then examined, and perfusion defects are seen as cold spots (low intensity regions) on the images. The patient is then subjected to a lung perfusion scan. The patient is injected with radioisotopelung boundaries than a simple threshold. How ever, a simple threshold combined with a median window filter provides almost the same result.

Figure 4.9 illustrates a ventilation and perfusion image pair from a person diagnosed as having an intermediate to high probability of pulmonary emboli (one in the upper left lobe and the other in the lower right lobe). This figure shows that the PCNN combined with a median window filter was able to highlight the lack of blood flow in the lower right lobe indicating a possible perfusion defect. 1-lowever, the PCNN processing of the ventilation image from the left lung makes it look as though there is an air obstruction in the upper left lobe. A trained physician would probably recognize that the shape was wrong and that

the perfusion image of the left lung was more likely indicating a perfusion defect only.

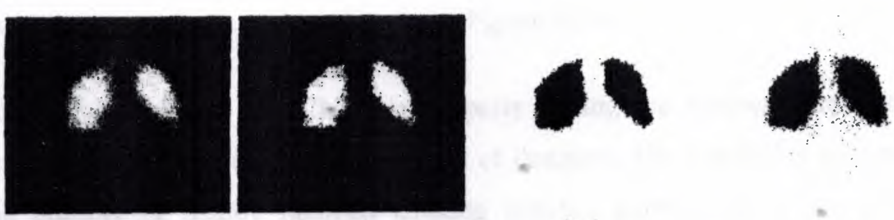

Figure 4.9. This figure shows an anterior perfusion image of a patient suffering chest pains but with no evidence of ventilation or perfusion problems (far left), the PCNN processed image (middle left), a PCNN segmentation of the lungs (middle right), and a simple thresholded image (far right).

processed image (middle left), a PCNN segmentation of the lungs (middle right), and a simple thresholded image (far right).

Figure 4.10. This figure shows a posterior ventilation and perfusion image pair (left) of a patient with a high probability of a pulmonary emboli in the upper left lobe and lower right lobe and the resulting image pair processed by a combination of PCNN and median window filter (right).

4.5 Results

The preliminary results show that PCNNs do well at contrast enhancement but require a good deal of manual intervention to produce the desired results. They also do well at image segmentation when each segment is approximately uniform in intensity. However, when intensity significantly varies across a single segment, that segment does not properly separate from other objects. MRI images have consistent contrast and low noise, which results in good segmentation as illustrated in Figure 4.3. However, VIQ scans are noisy images with varying contrast, which can result in improper segmentation as indicated by the results Figure 4.10.

Another complexity of PCNNs is properly setting the various parameters so that a uniform response is achieved over a set of imagery. The adjustable parameters include the number of nearby neurons (linking inputs), number of nearby pixels (feeding inputs), strengths of the linking and feeding connections, decay constants, and thresholds. For example, a set of parameters, which properly segment objects in one image, can fail on similar images. For our evaluation, we continually changed the parameters until we got the desired enhancement or segmentation.

Speed can be an issue for real-time image processing. Our evaluation was performed on a desktop computer (266 MHz Pentium II with 200 Mbytes of RAM and Windows NT).

The image sets used in this evaluation were 256 by 256 pixels in size. It generally took between 5 and 30 seconds to process a single image.

Finally, our examination of the image segmentation process verifies that PCNNs are incorporating both spatial and intensity values into the threshold process.

4.6 Summary

This investigation shows that PCNNs are useful for image segmentation and edge detection on imagery with low noise and with consistent contrast across the image. A PCNN is essentially an adaptive thresholding algorithm that incorporates local spatial information to the thresholding process. A PCNN can be used as an image-processing tool or as a preprocessor to a machine vision system. It has some problems with noisy images. Without the aid of specialized hardware, processing times are too slow for real-time image processing.

Overall, PCNNs have several benefits to image processing and are worth exploring, though an understanding of their limitations is necessary. Also, for PCNNs to gain popularity and find usefulness, a better approach to setting parameters is necessary, and a better understanding the parameter relationships should be researched.

CONCLUSION

In chapter one I described the definition of artificial neural network and provided brief history of neural networks. After that I have moved to give a hint about how artificial neuron works.

In chapter two I tried to show the algorithms and the structure of the neural network which are feed-forward network and feedback network.

In Chapter three I gave some applications of neural networks in medicine and described the integration of neural networks and knowledge based systems in medicine, and presented the clinical application.

In chapter four it is written about Pulse-Coupled Neural Networks and how its support the image analysis medically for human brain imaginary.

In Artificial neural network is being developed all the time especially in medicine it will be important to investigate and learn about neural network, then emergence, structure and application in real life.

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/tech/neural/neural_toc.html

[3] Academic writing from the World Wide Web:

http://www.usps.gov/websites/

[4] Jones, Stephen, Neural Networks and the Computational Brain or Matters Relating to Artificial Intelligence.

[5] Some specific models of artificial neural nets, Lecture Notes.

[6] Cariani, P. "Emergence and Artificial Life." In C. G. Langton, C. Taylor, J. D. Farmer and S. Rasmussen, eds., Artificial Life II. Addison-Wesley. (1991).

[7] O'Reilly, R. C., & McClelland, J. L., "The Self-Organisation of Spatially Invariant Representations." Technical Report PDP.CNS.92.5. Department of Psychology, CMU.

[8] Rochester, N., J. H. Holland, et al. "Tests on a Cell Assembly Theory of the Action of the Brain, Using a Large Digital Computer." In J.A. Anderson and E. Rosenfeld, eds., 1988, Neurocomputing. Cambridge, Mass.: MIT Press. (1956).

[9] Sima, J., and Orponen, P. "Computing with continuous-time Liapunov systems," 33rd ACM STOC, 2001

[10]Frasconi P., Gori M., and Sperduti A., ``A Framework for Adaptive Data Structures Processing", IEEE Transactions on Neural Networks. *Vol. 9, No. 5, pp. 768-786, 1998.*

[11]Frasconi P., Goller C., Gori M., Kuchler A., and Sperduti A., ``From Sequences to Data Structures: Theory and Applications'', in ``Dynamical Recurrent Networks'', Eds. J. Kolen and S.C. Kremer, IEEE Press.

[12] de Dombal F.T., Leaper D.J., Staniland J.R., McCann A.P., Horrocks J.C. *Acute Abdominal Pain*, Brit. Med. J. 2: 9-13, (1972).

[13]Schmidt, G. Thews. 1983. Physiologie des Menschen. Springer-Verlag. Berlin - Heidelberg - New York.

[14] Angel. 1997. Interactive Computer Graphics. Addison Wesley.

[15] Vander, J.H. Sherman, D.S. Luciano. 1990. Human Physiology: The Mechanisms of Body Function. McGraw-Hill

Publishing Company, New York.

[16] Tehrani. 1993. "Mathematical Analysis and Computer Simulation of the Respiratory System in the Newborn Infant."

IEEE Transactions on Biomedical Engineering, Vol. 40, No. 5, pp. 475-481.

[17]Fox PT. Basic Principles and Neurosurgical Applications of Positron Emission Tomography. Neurosurg Clin N Am 1997; 8: 293–306.

[18]Kohonen T: Self-organised formation of topologically correct feature maps. Biological Cybernetics, 1982; 43: 59-69

[19] Cresswell P. Assembly, transport, and function of MHC class II molecules. Annual Review in Immunology, 12, 259-293, (1994).

[20] Wang R., Doolan D.L., Le T.P., Hedstrom R.C., Coonan K.M., Charoenvit Y., Jones J.A. and Hoffman S.L. Induction of antigen-specific cytotoxic T lymphocytes in humans by a malaria DNA vaccine. Science, 282, 476–480, (1998).

[21] Brusic V., Rudy G. and Harrison L.C. MHCPEP – a database of MHC-binding peptides: update 1997. Nucleic Acids Research, 26, 368-371(1998).