**www.SeminarsTopics.com**

**TOPIC: ARTIFICIAL NEURAL NETWORK (ANN)**

1. **INTRODUCTION**

The simplest definition of a neural network, more properly referred to as an 'artificial' neural network (ANN), is provided by the inventor of one of the first neurocomputers, Dr. Robert Hecht-Nielsen. He defines a neural network as:

*"...a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs.* “

ANNs are processing devices (algorithms or actual hardware) that are loosely modeled after the neuronal structure of the mamalian cerebral cortex but on much smaller scales. A large ANN might have hundreds or thousands of processor units, whereas a mamalian brain has billions of neurons with a corresponding increase in magnitude of their overall interaction and emergent behavior. Although ANN researchers are generally not concerned with whether their networks accurately resemble biological systems, some have. For example, researchers have accurately simulated the function of the retina and modeled the eye rather well.

**1.2 BACKGROUND STUDY**

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system (Eric Davalo and Patrick Naim, 1986). It is composed of a large number of highly interconnected processing elements (neurones) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurones. This is true of ANNs as well.

**What is a neural network?**

In information technology, a neural network is a system of programs and data structures that approximates the operation of the human brain. A neural network usually involves a large number of processors operating in parallel, each with its own small sphere of knowledge and access to data in its local memory. Typically, a neural network is initially "trained" or fed large amounts of data and rules about data relationships (for example, "A grandfather is older than a person's father"). A program can then tell the network how to behave in response to an external stimulus (for example, to input from a computer user who is interacting with the network) or can initiate activity on its own (within the limits of its access to the external world) (Eric Davalo and Patrick Naim Assimov, 1984).

Neural Networks are a different paradigm for computing:

* von Neumann machines are based on the processing/memory abstraction of human information processing.
* neural networks are *based on the parallel architecture of animal brains*.

Neural networks are a form of multiprocessor computer system, with

* simple processing elements
* a high degree of interconnection
* simple scalar messages
* 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.

Example: A simple single unit adaptive network:

The network has 2 inputs, and one output. All are binary.

The output is

1 if W0 \*I0 + W1 \* I1 + Wb > 0

0 if W0 \*I0 + W1 \* I1 + Wb <= 0

We want it to learn simple OR: output a 1 if either I0 or I1 is 1.

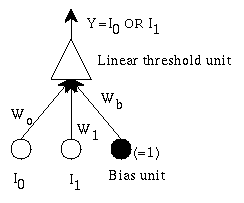


Figure 1: simple single unit adaptive network

**1.3 BENEFITS OF NEURAL NETWORK**

1. **Nonlinearity:** A neuron is basically a nonlinear device. Consequently, a neural network, made up of an interconnection of neurons, is itself nonlinear. Moreover, the nonlinearity is of a special kind in the sense that it is distributed throughout the network.

2. **Input-output mapping:** A popular paradigm of learning called *supervised learning* involves the modification of the synaptic weights of a neural network by applying a set oftraining samples. Each sample consists of a unique input signal and the correspondingdesired response (Fausett L., 1994). The network is presented a sample picked at random from the set, andthe synaptic weights (free parameters) of the network are modified so as to minimize thedifference between the desired response and the actual response of the network producedby the input signal in accordance with an appropriate criterion. The training of thenetwork is repeated for many samples in the set until the network reaches a steady state,where there are no further significant changes in the synaptic weights. The previouslyapplied training samples may be reapplied during the training session, usually in adifferent order. Thus the network learns from the samples by constructing an input outputmapping for the problem at hand.

**3.** **Adaptivity:** Neural networks have a built-in capability to adapt their synaptic weights to changes in the surrounding environment. In particular, a neural network trained to operate in a specific environment can be easily retrained to deal with minor changes in the operating environmental conditions. Moreover, when it is operating in a nonstationary environment a neural network can be designed to change its synaptic weights in real time. The natural architecture of a neural network for pattern classification, signal processing, and control applications, coupled with the adaptive capability of the network, makes it an ideal tool for use in adaptive pattern classification, adaptive signal processing, and adaptive control.

**4. Contextual information:** Knowledge is represented by the very structure and activation state of a neural network. Every neuron in the network is potentially affected by the global activity of all other neurons in the network. Consequently, contextual information is dealt with naturally by a neural network.

**5. Fault tolerance:** A neural network, implemented in hardware form, has the potential to be inherently fault tolerant in the sense that its *performance is degraded gracefully* under adverse operating. For example, if a neuron or its connecting links are damaged, recall of a stored pattern is impaired in quality. However, owing to the distributed nature of information in the network, the damage has to be extensive before the overall response of the network is degraded seriously. Thus, in principle, a neural network exhibits a graceful degradation in performance rather than catastrophic failure.

**6. VLSI implementability:** The massively parallel nature of a neural network makes it potentially fast for the computation of certain tasks. This same feature makes a neural network ideally suited for implementation using very-large-scale-integrated (VLS1) technology.

**7. Uniformity of analysis and design:** Basically, neural networks enjoy universality as information processors. We say this in the sense that the same notation is used in all the domains involving the application of neural networks. This feature manifests itself in different ways:

a) Neurons, in one form or another, represent an ingredient *common* to all neural networks.

b) This commonality makes it possible to *share* theories and learning algorithms in different applications of neural networks.

c) Modular networks can be built through a *seamless integration of modules*.

**8. Neurobiological analogy:** The design of a neural network is motivated by analogy with the brain, which is a living proof that fault-tolerant parallel processing is not only physically possible but also fast and powerful. *Neurobiologists* look to (artificial) neural networks as a research tool for the interpretation of neurobiological phenomena. On the other hand, *engineers* look to neurobiology for new ideas to solve problems more complex than those based on conventional hard-wired design techniques.

The neurobiological analogy is also useful in another important way: It provides a hope and belief that physical understanding of neurobiological structures could influence the art of electronics and thus VLSI.

**1.4 OBJECTIVE STUDY**

The purpose of this study is to outline the reason why we use Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, and how it can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques.

**1.5 SCOPE STUDY**

The scope of this research work is based on how the brain trains itself to process information. 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.

**1.6 LIMITATION**

There are some specific issues potential users should be aware of.

* Backpropagational neural networks (and many other types of networks) are in a sense the ultimate 'black boxes'. Apart from defining the general archetecture of a network and perhaps initially seeding it with a random numbers, the user has no other role than to feed it input and watch it train and await the output. In fact, it has been said that with backpropagation, "you almost don't know what you're doing". Some software freely available software packages (NevProp, bp, Mactivation) do allow the user to sample the networks 'progress' at regular time intervals, but the learning itself progresses on its own. The final product of this activity is a trained network that provides no equations or coefficients defining a relationship (as in regression) beyond it's own internal mathematics. The network 'IS' the final equation of the relationship (Freeman J.A., Skapura D.M, 1991).
* Backpropagational networks also tend to be slower to train than other types of networks and sometimes require thousands of epochs. If run on a truly parallel computer system this issue is not really a problem, but if the BPNN is being simulated on standard serial machine (i.e. a single SPARC, Mac or PC) training can take some time. This is because the machines CPU must compute the function of each node and connection separately, which can be problematic in very large networks with a large amount of data. However, the speed of most current machines is such that this is typically not much of an issue.

**CHAPTER TWO**

1. **LITERATURE REVIEW**

**2.1** **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*, 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 neurones (Zurada J.M., 1992). 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.

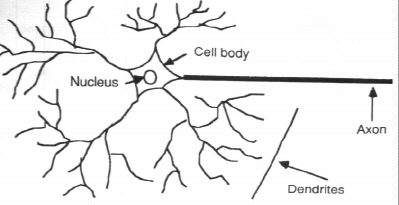


Figure 2: Components of a neuron

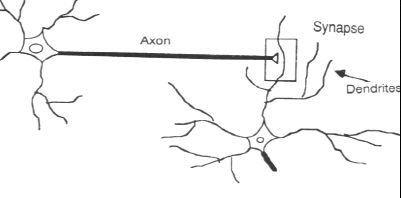
****

Figure 3: The synapse

**2.2** **From Human Neurones to Artificial Neurones**

We conduct these neural networks by first trying to deduce the essential features of neurones and their interconnections. We then typically program a computer to simulate these features. However because our knowledge of neurones is incomplete and our computing power is limited, our models are necessarily gross idealisations of real networks of neurones (Duda R.O., Hart P.E., Stork D.G, 2001).

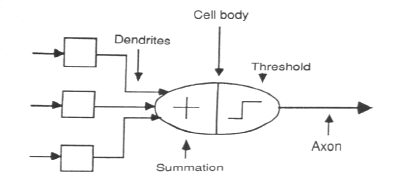


Figure 4: The neuron model

**CHAPTER THREE**

1. **RESEARCH METHODLOGY**

**3.1 A simple neuron**

An artificial neuron is a device with many inputs and one output. The neuron has two modes of operation; the training mode and the using mode. In the training mode, the neuron can be trained to fire (or not), for particular input patterns. In the using mode, when a taught input pattern is detected at the input, its associated output becomes the current output. If the input pattern does not belong in the taught list of input patterns, the firing rule is used to determine whether to fire or not.

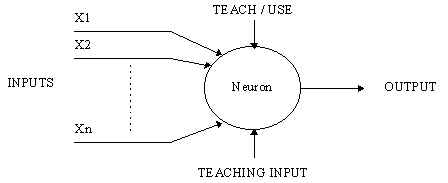


Figure 5: A simple neuron

**3.2 Firing rules**

The firing rule is an important concept in neural networks and accounts for their high flexibility. A firing rule determines how one calculates whether a neuron should fire for any input pattern. It relates to all the input patterns, not only the ones on which the node was trained (Russell S., Norvig P., 2003).

A simple firing rule can be implemented by using Hamming distance technique. The rule goes as follows:

Take a collection of training patterns for a node, some of which cause it to fire (the 1-taught set of patterns) and others which prevent it from doing so (the 0-taught set). Then the patterns not in the collection cause the node to fire if, on comparison , they have more input elements in common with the 'nearest' pattern in the 1-taught set than with the 'nearest' pattern in the 0-taught set. If there is a tie, then the pattern remains in the undefined state.

For example, a 3-input neuron is taught to output 1 when the input (X1,X2 and X3) is 111 or 101 and to output 0 when the input is 000 or 001. Then, before applying the firing rule, the truth table is;

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X1: |  | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| X2: |  | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| X3: |  | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
|  |  |  |  |  |  |  |  |  |  |
| OUT: |  | 0 | 0 | 0/1 | 0/1 | 0/1 | 1 | 0/1 | 1 |

As an example of the way the firing rule is applied, take the pattern 010. It differs from 000 in 1 element, from 001 in 2 elements, from 101 in 3 elements and from 111 in 2 elements. Therefore, the 'nearest' pattern is 000 which belongs in the 0-taught set. Thus the firing rule requires that the neuron should not fire when the input is 001. On the other hand, 011 is equally distant from two taught patterns that have different outputs and thus the output stays undefined (0/1).

By applying the firing in every column the following truth table is obtained;

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X1: |  | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| X2: |  | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| X3: |  | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
|  |  |  |  |  |  |  |  |  |  |
| OUT: |  | 0 | 0 | 0 | 0/1 | 0/1 | 1 | 1 | 1 |

The difference between the two truth tables is called the *generalisation of the neuron.* Therefore the firing rule gives the neuron a sense of similarity and enables it to respond 'sensibly' to patterns not seen during training.

**3.3** **Pattern Recognition - an example**

An important application of neural networks is pattern recognition. Pattern recognition can be implemented by using a feed-forward (figure 6) neural network that has been trained accordingly Principe (J.C., Euliano N.R., Lefebvre W.C., 2000). During training, the network is trained to associate outputs with input patterns. When the network is used, it identifies the input pattern and tries to output the associated output pattern. The power of neural networks comes to life when a pattern that has no output associated with it, is given as an input. In this case, the network gives the output that corresponds to a taught input pattern that is least different from the given pattern.

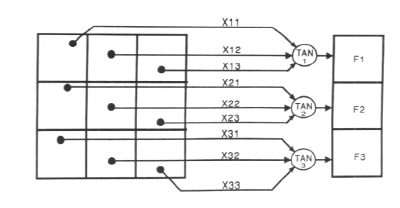


Figure 6: Pattern Recognition

For example:

The network of figure 6 is trained to recognise the patterns T and H. The associated patterns are all black and all white respectively as shown below.



If we represent black squares with 0 and white squares with 1 then the truth tables for the 3 neurones after generalisation are;

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X11: |  | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| X12: |  | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| X13: |  | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
|  |  |  |  |  |  |  |  |  |  |
| OUT: |  | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |

**Top neuron**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X21: |  | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| X22: |  | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| X23: |  | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
|  |  |  |  |  |  |  |  |  |  |
| OUT: |  | 1 | 0/1 | 1 | 0/1 | 0/1 | 0 | 0/1 | 0 |

**Middle neuron**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X21: |  | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| X22: |  | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| X23: |  | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
|  |  |  |  |  |  |  |  |  |  |
| OUT: |  | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |

**Bottom neuron**

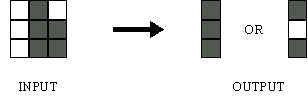
 From the tables it can be seen the following associasions can be extracted:



In this case, it is obvious that the output should be all blacks since the input pattern is almost the same as the 'T' pattern.



Here also, it is obvious that the output should be all whites since the input pattern is almost the same as the 'H' pattern.



Here, the top row is 2 errors away from the a T and 3 from an H. So the top output is black. The middle row is 1 error away from both T and H so the output is random. The bottom row is 1 error away from T and 2 away from H. Therefore the output is black. The total output of the network is still in favour of the T shape.

**3.4** **A more complicated neuron**

The previous neuron doesn't do anything that conventional computers don't do already. A more sophisticated neuron (figure 7) is the McCulloch and Pitts model (MCP). The difference from the previous model is that the inputs are ‘weighted’; the effect that each input has at decision making is dependent on the weight of the particular input. The weight of an input is a number which when multiplied with the input gives the weighted input. These weighted inputs are then added together and if they exceed a pre-set threshold value, the neuron fires. In any other case the neuron does not fire (Hertz J., Krogh A., Palmer R.G, 1991).

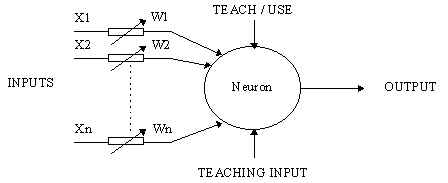


Figure 7. An MCP neuron

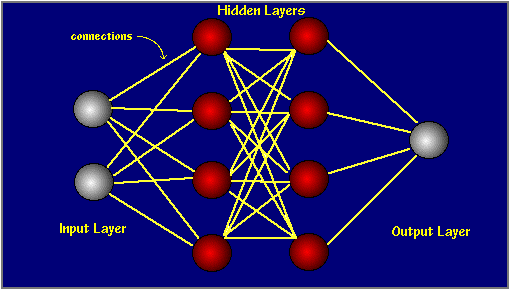
In mathematical terms, the neuron fires if and only if;

X1W1 + X2W2 + X3W3 + ... > T

The addition of input weights and of the threshold makes this neuron a very flexible and powerful one. The MCP neuron has the ability to adapt to a particular situation by changing its weights and/or threshold. Various algorithms exist that cause the neuron to 'adapt'; the most used ones are the Delta rule and the back error propagation. The former is used in feed-forward networks and the latter in feedback networks.

## 3.1 The Basics of Neural Networks

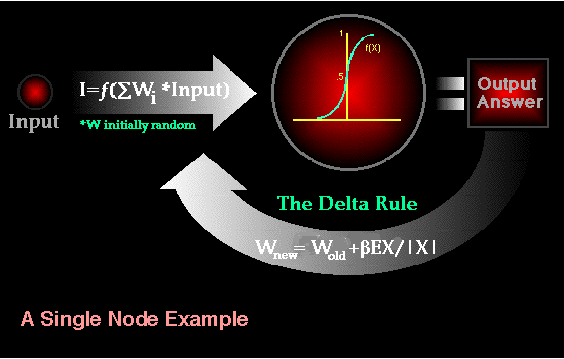
Neural neworks are typically organized in layers. Layers are made up of a number of interconnected 'nodes' which contain an 'activation function'. Patterns are presented to the network via the 'input layer', which communicates to one or more 'hidden layers' where the actual processing is done via a system of weighted 'connections'. The hidden layers then link to an 'output layer' where the answer is output as shown in the graphic below (Haykin S., 1999).



Most ANNs contain some form of 'learning rule' which modifies the weights of the connections according to the input patterns that it is presented with. In a sense, ANNs learn by example as do their biological counterparts; a child learns to recognize dogs from examples of dogs.

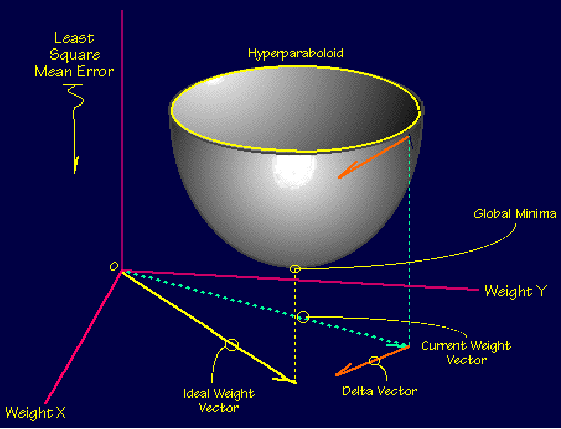
Although there are many different kinds of learning rules used by neural networks, this demonstration is concerned only with one; the delta rule. The delta rule is often utilized by the most common class of ANNs called 'backpropagational neural networks' (BPNNs). Backpropagation is an abbreviation for the backwards propagation of error.

With the delta rule, as with other types of backpropagation, 'learning' is a supervised process that occurs with each cycle or 'epoch' (i.e. each time the network is presented with a new input pattern) through a forward activation flow of outputs, and the backwards error propagation of weight adjustments. More simply, when a neural network is initially presented with a pattern it makes a random 'guess' as to what it might be. It then sees how far its answer was from the actual one and makes an appropriate adjustment to its connection weights. More graphically, the process looks something like this:



Note also, that within each hidden layer node is a sigmoidal activation function which polarizes network activity and helps it to stablize.

Backpropagation performs a gradient descent within the solution's vector space towards a 'global minimum' along the steepest vector of the error surface. The global minimum is that theoretical solution with the lowest possible error. The error surface itself is a hyperparaboloid but is seldom 'smooth' as is depicted in the graphic below. Indeed, in most problems, the solution space is quite irregular with numerous 'pits' and 'hills' which may cause the network to settle down in a 'local minum' which is not the best overall solution.



Since the nature of the error space cannot be known a prior, neural network analysis often requires a large number of individual runs to determine the best solution. Most learning rules have built-in mathematical terms to assist in this process which control the 'speed' (Beta-coefficient) and the 'momentum' of the learning. The speed of learning is actually the rate of convergence between the current solution and the global minimum. Momentum helps the network to overcome obstacles (local minima) in the error surface and settle down at or near the global miniumum.

Once a neural network is 'trained' to a satisfactory level it may be used as an analytical tool on other data. To do this, the user no longer specifies any training runs and instead allows the network to work in forward propagation mode only. New inputs are presented to the input pattern where they filter into and are processed by the middle layers as though training were taking place, however, at this point the output is retained and no backpropagation occurs. The output of a forward propagation run is the predicted model for the data which can then be used for further analysis and interpretation.

It is also possible to over-train a neural network, which means that the network has been trained exactly to respond to only one type of input; which is much like rote memorization. If this should happen then learning can no longer occur and the network is refered to as having been "grandmothered" in neural network jargon. In real-world applications this situation is not very useful since one would need a separate grandmothered network for each new kind of input.

## 3.2 How Do Neural Networks Differ From Conventional Computing?

To better understand artificial neural computing it is important to know first how a conventional 'serial' computer and it's software process information (Klimasauskas, CC., 1989). A serial computer has a central processor that can address an array of memory locations where data and instructions are stored. Computations are made by the processor reading an instruction as well as any data the instruction requires from memory addresses, the instruction is then executed and the results are saved in a specified memory location as required. In a serial system (and a standard parallel one as well) the computational steps are deterministic, sequential and logical, and the state of a given variable can be tracked from one operation to another.

In comparison, ANNs are not sequential or necessarily deterministic. There are no complex central processors, rather there are many simple ones which generally do nothing more than take the weighted sum of their inputs from other processors. ANNs do not execute programed instructions; they respond in parallel (either simulated or actual) to the pattern of inputs presented to it. There are also no separate memory addresses for storing data. Instead, information is contained in the overall activation 'state' of the network. 'Knowledge' is thus represented by the network itself, which is quite literally more than the sum of its individual components.

**3.3 APPLICATIONS OF NEURAL NETWORK**

**3.3.1** **Neural networks in medicine**

Artificial Neural Networks (ANN) are currently a 'hot' research area in medicine and it is believed that they will receive extensive application to biomedical systems in the next few years Principe (J.C., Euliano N.R., Lefebvre W.C., 2000). At the moment, the research is mostly on modelling parts of the human body and recognising diseases from various scans (e.g. cardiograms, CAT scans, ultrasonic scans, etc.).

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

**3.3.2 Modelling and Diagnosing the Cardiovascular System**

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

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

Another reason that justifies the use of ANN technology, is the ability of ANNs to provide 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 analysed. In medical modelling and diagnosis, this implies that even though each sensor in a set may be sensitive only to a specific physiological variable, ANNs are capable of detecting complex medical conditions by fusing the data from the individual biomedical sensors.

**3.3.3 Electronic noses**

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

**3.3.4 Instant Physician**

An application developed in the mid-1980s called the "instant physician" trained an autoassociative memory neural network to store a large number of medical records, each of which includes information on symptoms, diagnosis, and treatment for a particular case. After training, the net can be presented with input consisting of a set of symptoms; it will then find the full stored pattern that represents the "best" diagnosis and treatment.

**3.3.5** **Neural Networks in business**

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

**3.3.6** **Marketing**

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

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

**3.3.7** **Credit Evaluation**

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

## 3.4 What Applications Should Neural Networks Be Used For?

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

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

**3.5 CONCLUSION/SUGGESTION**

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

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

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

Finally, I would like to state that even though neural networks have a huge potential we will only get the best of them when they are integrated with computing.

**REFERENCE**

Eric Davalo and Patrick Naim, Learning internal representations by error propagation by Rumelhart, Hinton and Williams (1986).

Eric Davalo and Patrick Naim Assimov, I (1984, 1950), Robot, Ballatine, New York. Electronic Noses for Telemedicine

Fausett L., *Fundamentals of Neural Networks*, Prentice-Hall, 1994. ISBN 0 13 042250 9 or

Freeman J.A., Skapura D.M.: *Neural networks - Algorithms, applications, and programming techniques*, Addison-Wesley, Reading, MA 1991

Gurney K., *An Introduction to Neural Networks*, UCL Press, 1997, ISBN 1 85728 503 4

Haykin S., *Neural Networks* , 2nd Edition, Prentice Hall, 1999, ISBN 0 13 273350 1 is a more detailed book, with excellent coverage of the whole subject.

Hertz J., Krogh A., Palmer R.G.: *Introduction to the theory of neural computation Addison Wesley*, Redwood City, CA 1991

Haykin S.: *Neural networks – A comprehensive Foundation* (2nd ed.), Prentice Hall, Upper Saddle River, 1999

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

Principe J.C., Euliano N.R., Lefebvre W.C.: *Neural and adaptive systems Fundamentals through simulations*, Wiley, New York, 2000

Duda R.O., Hart P.E., Stork D.G.: *Pattern classification*, Wiley, New York, 2001

Russell S., Norvig P.: *Artificial intelligence – A modern approach* (2nd ed.), Prentice Hall, Upper Saddle River, 2003

Zurada J.M.: *Introduction to artificial neural systems*, West Publishing Company, St. Paul 1992