ARTIFICIAL NEURAL NETWORKS IN ENGINEERING EDUCATION

Terje Kristensen[1]

Abstract - *The human brain and related concepts such as learning, knowledge and computing are discussed. The Artificial Neural Network (ANN) paradigm is introduced as a model of computing. Some aspects of the brain and a neural network such as knowledge representation are discussed. Some small prototypes of artificial neural networks systems, like a pattern recognition system, an associative memory system and a natural language processing (NLP) system are introduced. All the ANN systems are simulated on a PC-platform.*

Keyword - *The brain, learning, knowledge, computing, ANN systems, simulation*

INTRODUCTION

The ability to learn is an important part of human intelligence. The brain's ability to learn and adapt makes it unique in nature. New data system try to imitate the way the brain solves problems by *learning* to perform the different tasks. The knowledge of the brain is represented in a *distributive* manner. One tries to imitate the way the brain both codes and processes knowledge by building artificial neural networks (ANN) that have built-in mechanisms for learning in the system.

The architecture of such systems is *distributive*, and ANN consists of many small units that are connected into a large network. The main difference between neural networks and conventional computer systems is the massive parallelism and redundancy they exploit in order to deal with the unreliability of the computing units.

THE HUMAN BRAIN

The human brain is in many ways the most complex object in our universe. The nervous system of the brain is an information processing totality. Signals from the environment are coded and processed to evoke an appropriate response. Moreover, biological networks are *self-organising* systems, and each neuron itself is also a self-organising structure capable of processing information in many different ways. The human nervous system is composed of millions of interconnected nerve cells or *neurons* where each of them is a very complex arrangement dealing with incoming signals in many different ways.

Massive and hierarchical networking of the brain seems to be the fundamental precondition for emergency of consciousness and complex behaviour. So far, biologists and neurologists have concentrated to uncover the properties of the individual neurons. The mechanism of production of transport signals from one neuron to the next is a well-understood physiological phenomenon, but how these individual systems co-operate to form complex and massive parallel systems, capable to carry out incredible complex information tasks, is far from being understood.

NEURAL NETWORKS

Artificial neural networks are an attempt at modelling the information processing capabilities of the nervous system. Neurons are slow compared to electronic gates. These can achieve switching times of a few nanoseconds, whereas neurons need several milliseconds to react to a stimulus. Nevertheless, the brain is capable of solving problems which no digital computer can yet efficiently deal with.

An artificial neural network is a system where the structure is only partially determined. Several parameters in the systems are not yet determined to give the solution of the problem. It is our task to determine these parameters. The adjustment of the parameters will be done through a *learning algorithm*. This is not through explicit programming, but more through an automatic adaptive programming method.

The interplay between the electrical transmission of information in the cell and chemical transmission between cells is the basis for neural information processing. Cells process information by integrating incoming signals and by reacting to inhibition. From this an *abstract neuron* can be modelled which contains *dendrites*, a *cell body* and an

[1] Terje Kristensen, The Department of Computer Science, Bergen University College, Nygårdsgaten 112, N-5020 Bergen , Norway. E-mail: tkr@hib.no.

axon. The dendrites receive signals from the other neurons, the cell body provide the supports function of the cell and the axon is the output mechanism for the neuron that conducts signals away from the cell to the other neurons via the interconnections points, the *synapses*. The same components will also be present in the artificial computing neuron.

Each processing unit of an ANN has multiple input arriving from the other units. Each connection has associated a *strength* (weight), given as $w_1, w_2, \ldots, w_n$. The processing unit performs a weighted sum on the inputs to compute its output. This output is sent to all the the output connections of this unit. In a typical network organisation of a neural network paradigm known as *backpropagation*, the network has different *layers* of processing units. The first layer is the input layer, the middle layer or the "hidden" layer, consists of "feature detectors"- units that respond to particular features that may appear in the input pattern. Sometimes there is more than one hidden layer. The last layer is the output layer. The activities of these units are read as output of the network. In some classifications, the output units stand for different classification of the patterns.

Neural networks are not programmed; they learn by example. The connection strenghts that are associated with each interconnection are adjusted during the training or the learning of the network. Most networks undergo a learning procedure during which the network weights are adjusted. The learning procedure may be *supervised* in which case the network is presented with target answers for each pattern that is input. In some network the learning algorithm is *unsupervised* – the network adjusts its weights in response to input patterns without the benefit of target answers. In unsupervised learning the network classifies the input patterns into similarity categories.

LEARNING, KNOWLEDGE AND COMPUTING

Learning

Human learning is a very complex phenomenon and consists of aspects like acquisition of knowledge, development of strategies to solve problems and ability to learn new patterns. Man has the ability to acquire knowledge in many areas and may learn in many ways. He has the ability to acquire both *declarative* and *procedural* knowledge. By declarative knowledge we mean facts and rules and by procedural knowledge we mean *algorithmic* knowledge. The first type may be characterised by the word *what* and the second type by the word *how*. To be more concrete, a *grammar* is for instance a *declarative* description of a language. *Recognition* on the other hand is a process that describes *how* to determine whether a particular string is a valid expression or not based upon the grammatical description of the actual language.

Knowledge

How do we know something? Knowledge is a very strange concept. It is difficult to discuss it because we need knowledge to do so. Knowledge is therefore a curious *self-referential* or *recursive* concept. To explain what knowledge really is, is very difficult, but how it may be described is a far more interesting aspect. Knowledge has to be *represented* in some way. This is a very important point whether we speak about the brain or a computer system. Knowledge is represented in many different media, like books, disks, CD, film and video. It is easier to describe knowledge by use of processes than by use of properties. Knowledge may in a way be saved, moved, retrieved and processed like any other material object we find in the world, but at the same time knowledge is *abstract* and *non-material*.

Humans express knowledge by use of their language. The natural language is only one way to represent knowledge, and not always the ideal one. Humans communicate and understand each other by use of the language. We also would like computers to process natural language. Automatic natural language processing is therefore an active research area in computing to day.

Knowledge has structure, but a fundamental problem is whether we can structure it without knowing how it is to be used. This is like finding a convincing argument for the existence of a universe without any one observing it. Equivalently, it is difficult to find a convincing argument for the existence of knowledge without a consciousness or brain that wants to use it.

Information and Computing

Knowledge is a self-referential concept and difficult to define. Information on the other hand may be defined more precisely like Claude Shannon did it in 1945. He defined information by use of *redundancy*. When information is represented, the *coding* of data is an important aspect. Coding plays an important role in information processing because when noise is present, information can still be transmitted without loss, if the right code with the right amount of redundancy is chosen.

In his classical paper from 1936 "*On Computable Numbers with an Application to the Entscheidungsproblem*" Alan Turing introduced the concept of *computability*. In his classical model computability was defined by a *Turing machine*, that is composed of an infinite tape, in which symbols can be stored and read. A read-write head can be moved to the left and to the right according to its internal state and is updated at each step. The *Turing thesis* states that *computable* functions are those which can be computed with this kind of device. The Turing approach made clear for the first time what "programming" really means, curiously at a time when no computer had yet been built.

An artificial neural network may be considered to be another approach to the problem of computation. We may ask what information capabilities such primitive computing units may have? What can really be computed with these networks. How can the networks determine their structure in a self-organising manner? In biological neural networks the information is stored at the contact points between different neurons, in their *synapses.* The neurons themselves are complex self-organising systems of signalling. Synapses determine the direction of transmission of information. Signals flow from one neuron to another in a well-defined manner. A well-defined direction of information is a basic element in every computing model.

A MEDICAL CLASSIFICATION SYSTEM

Systems that may identify and classify two-dimensional images based on their shapes are very much needed in many software applications. In figure 1 a three layered Backpropagation network is used for such a study. Both the input and the output layer is arranged as two-dimensional grids (7x9) and the hidden layer has 12 processing units.

The training set in this case consists of four medical images that have been simplified in the pattern recognition task. The four images are based on shapes of different one-cell organisms. The organisms may be amoeboid cells, distinctable by the thickness of their membrane and the organising of their nucleus. We see one training of the patterns in figure 1. The training set is a considerable simplification of real life images that may be scanned from real cell tissues. The training set was presented to the network about 1000 times, and the network was then able to classify all the images in the training set. In addition good performance was achieved on a test with noisy pictures.

The target outputs of the neural network are given by a vertical bar of activity. The bar is to appear at a different place based on the network's classification of the cell type. There are in this experiment four different positions of the bar, corresponding to the four different cell types. This is only one way of coding the target pattern and is done more for pedagogical reasons. To classify four different cell types only two bits are needed. A screen dump of the training session of the neural netwok is illustrated in figure 2. In this case only two hidden units have been used.

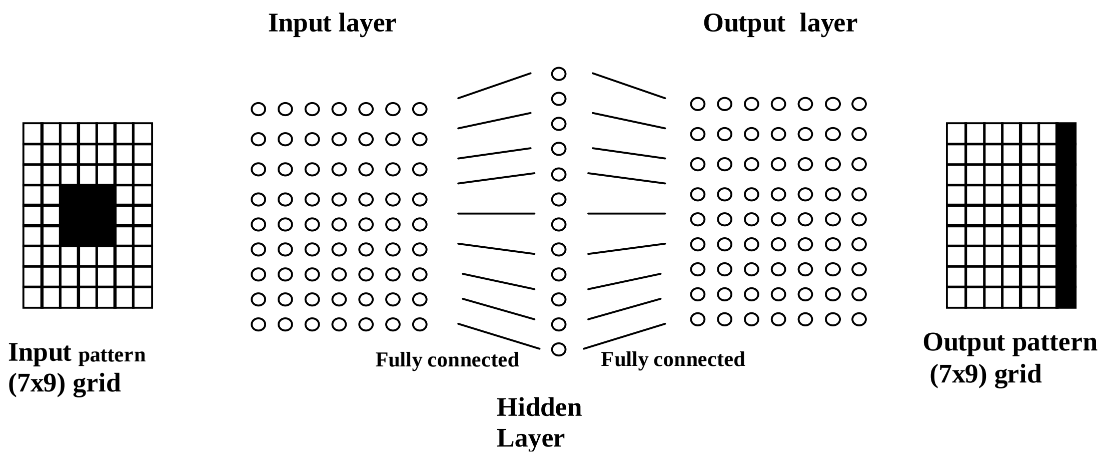

Figure 1. A three layered Backpropagation network may be trained to classify different cell types.

AN ASSOCIATIVE MEMORY SYSTEM

A quite different neural network architecture is a *Hopfield* network. A binary Hopfield network has only one single layer of processing elements. Each processing unit has an activity or state that is binary. The entire network is considered to have a state at each moment. The state is given by a vector of 0s and 1s. The processing elements are fully interconnected - each unit is connected to every other unit. Each connection has a strength which is denoted a weight. The weight w_{ji} is the strength of the connection from unit i to unit j.

Figure 2. A screen dump of the classification system of different cell types.

In the Hopfield network we assume that w_{ji} is equal to w_{ij} – the weights are *symmetrical*. It can be shown that under this condition the network is able to converge and eventually attains a stable state.

Initially, the network is assigned a state for each processing node. One unit is updated at a time. The updating procedure affects the state of each unit, sometimes changing it and sometimes leaving it the same. In the updating procedure each neuron takes a weighted sum of its inputs, that is $S_j = \sum a_i w_{ji}$. Each unit evaluates this sum. When the sum is greater or equals 0 the output of the node is set to 1, and if it is less than 0 the output is set to 0.

One method for updating is to update in sequence and repeat the process until a stable state is attained. However, this was not the updating process chosen by Hopfield. He chose an updating mechanism where the next unit to be updated was picked at *random*. In this way all units were allowed to have the same average updating rate. Usually many updates must be done before the network reaches a stable state. The random updating process in the Hopfield network provides an important difference between this neural network and other paradigms. If we compare this regime with what is happening in the brain, such asynchronous updating is closer to biological reality. In the brain the neurons update their states due to events impinging upon the neuron. The events are not synchronised for the neurons in the brain.

In an associative memory each "memory" is represented by a vector. Each memory vector is a state that corresponds to a *minimum* value of a network *energy* or *cost* function. The network starts in an initial position, and the updating procedure move the network until it is in one of the "memories" of the network. We may say that the network *converges* to the memory state that is accessible from the initial state. Often one calls such a state an *attractor*. The network is expected to converge to the memory that is most accessible to its initial state. If the pattern is noisy, the network will still converge to the correct state. The correct data would then in principle be recovered in spite of noise in the data.

Unfortunately, a Hopfield network has many limitations to be put into real use. First, the evoked memory in not necessarily the pattern that is most similar to the input pattern. Second, sometimes the network evokes so-called *spurious* states - patterns that were not stored in the memory. There is also a maximum limit to the number of memories that can be stored in a Hopfield network. This number increases when the number of nodes increases. The total number of memories is 0.15N, where N is the number of processing nodes in the network. The result has been found by experimenting with simulations. If the result is applied to the brain and the number of neurons here set equal to $N = 10^{11}$, the memory capacity of the brain is calculated to about 10^{10} memory patterns – a number that may not be too far from reality.

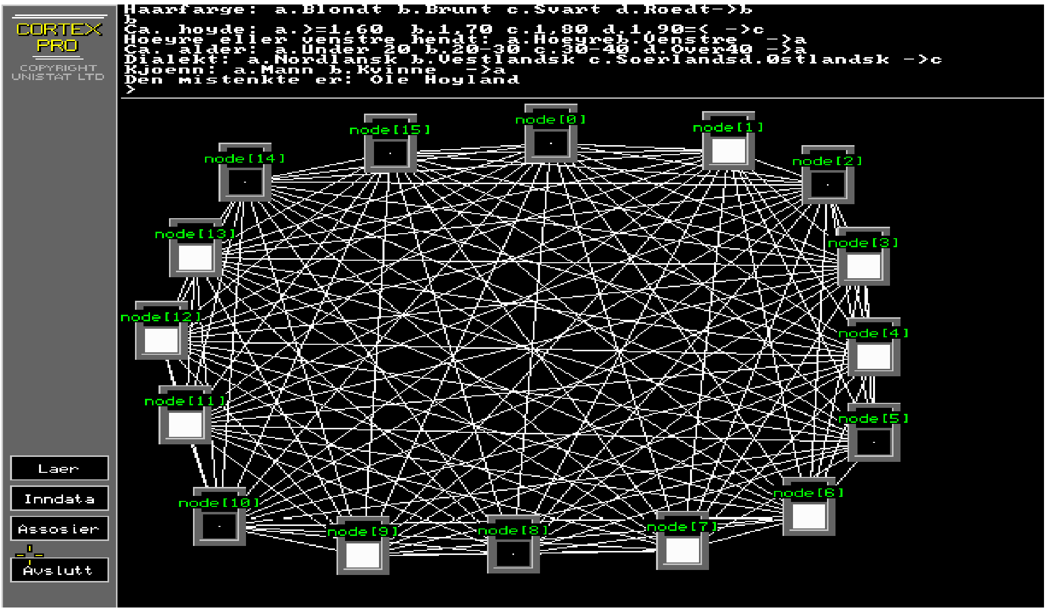

Figure 3. A Hopfield network illustrating an associative memory system.

In figure 3 we have used a Hopfield network to build a practical associative memory system. Our associative memory consists in this case of an archive of criminals in a police register with different attributes like eye colour, hair type, height, sex, etc. Each memory vector consists of 16 bits, 12 bits or neurons to code the different attributes and four bits to code the names of each criminal. Four bits are used to map between the attributes of the criminals and their names. In this case each node is updating in sequence. We are letting the updating procedure iterate 10000 iterations, and we will assume that this number will be great enough to stabilise the network. A possible application of this memory system may be that a witness has for instance seen a person and has noticed some of his attributes like sex, hair type, height, etc. These partial data should in principle be sufficient for the network to recall the right person.

A NATURAL LANGUAGE PROCESSING APPLICATION

In this section we are going to study a *counterpropagation* network and see how this network architecture may be applied in a natural language processing application. Counterpropagation combines two different layers from other paradigms and construct a new type of network that may be suitable to handle recogniton of different characters of a language. The network architecture consists of 3 layers: the input layer, the hidden layer that is a Kohonen layer with competitive units that do unsupervised learning and the top layer that is a Grossberg layer. The top layer is not competitive. The Grossberg layer is trained by a Widrow-Hoff learning rule [1].

First, an input pattern is presented to the network. The nodes of the hidden layer sum the inputs and compete to best respond to the input pattern. A single unit wins the competition and becomes activated. The remaining nodes become inactive. The winning node represents the classification category of the input pattern. The winning node's activation level is set to 1 and the activation level of the other hidden units are set to 0. The unit that wins the competition activates a pattern in the top layer. The winning unit is fully connected to the top layer and each connection has a weight.

In a Counterpropagation network there are two layers of weights represented by two vectors. The first one represents the weights incoming to a competetive unit j and the second one from the competitive unit to an output unit k. The two vectors often also have to satisfy some *normalisation* conditions. During training both layers are updated. The first after a competitive winner is chosen. The competitive winner is chosen in response to an input pattern. The network's output values are compared to the target pattern and the second layer of weights is then updated. Each layer also have a learning constant that is usually chosen between 0 and 1.

A number of variations can be made to the standard Counterpropagation training scheme. The network may be trained in two steps. In the first step only bottom weights of the bottom layer are adjusted. In the second training step the second layer of weights are adjusted alone. The reason for this training scheme is that the second layer' s weights do not develop meaningful weights before the first layer has undergone some training. In the natural language application in figure 4 this kind of training scheme has been used.

Recognition of characters

To demonstrate how the counterpropagation network works, the network is trained to recognise some characters. Each character is represented in a 5x5 grids structure in the input layer as shown in figure 4. In the next layer the input values are transformed to satisfy a normalisation condition. The third layer in figure 4 is the competitive layer which is a Kohonen layer. The fourth layer is the Grossberg layer that is used to recognise the different characters.

The program also contains a test panel to test the different characters that the network has been trained on. We may also set the different values of the nodes of the input layer (0 or 1) in an *interactive* way to see how the network categorises the character. The last window in figure 4 is not part of the neural network, but is a separate graphical window that is used to interpret the result of the Grossberg layer to see if the character is one of the characters the network has been trained on.

We may also introduce noise in the input patterns which are presented to the network and see how much the input patterns may be disturbed before the network is not able to recognise the character.

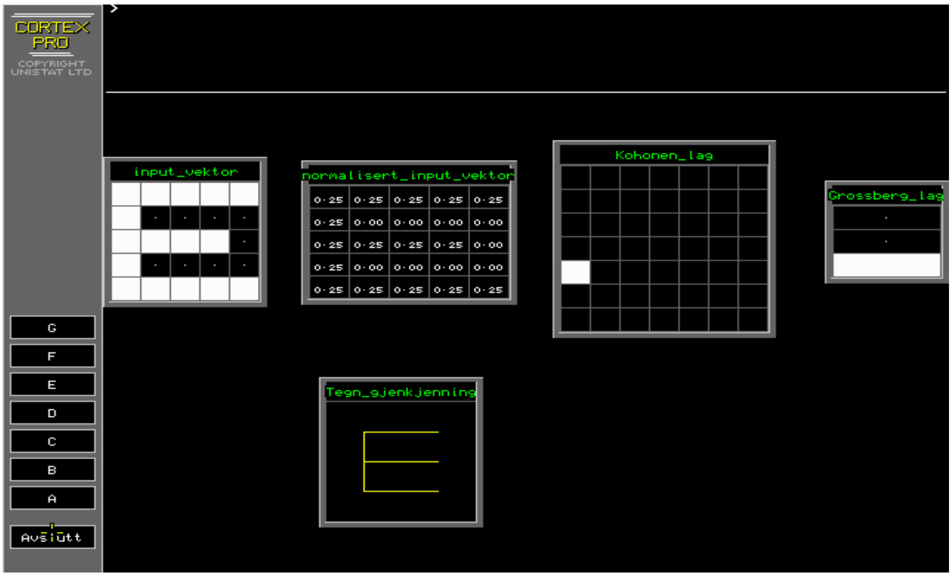

Figure 4. A simple demonstration of an ANN natural language application.

CONCLUSION

The human brain and related concepts such as learning, knowledge and computing have been discussed in the paper. ANN has been introduced as a model of computing and related to the concept of a Turing machine. In fact, it can be shown that these two models of computing are equivalent.

Some small prototypes of neural network applications have been developed to show how different paradigms such as Backpropagation, Hopfield and Counterpropagation may be used in real applications. A course in ANN should be introduced with emphasis on the practical aspects of ANN. It is quite essential that the students develop their own models and implement them. The students should experiment with the models to see how they function in practice. The more theoretical aspects of ANN, combined with advanced applications [2,3,4] should be taught when the students have acquired a solid understanding of the different subjects.

REFERENCES

[1] HayKin,S. Neural Networks. A comprehensive foundation. Prentice Hall , Ontario, Canada, 1999.

[2] Kristensen,T., Langmyhr,D. Two Approaches to Hyphenating Norwegian. In Proceedings of the Norwegian Conference of Informatics, NIK'98. Tapir Publisher, Kristiansand, Norway, 1998.

[3] Kristensen, T. A Neural Network Approach to Hyphenating Norwegian. In Proceedings of I.E.E.E International Joint Conference of Neural Network, IJCNN'2000, volume 2, Como, Italy, 2000.

[4] Kristensen,T., Treck,B. A Norwegian NETtalk Program. In Proceedings of the Norwegian Conference of Informatics, NIK'2000, Tapir Publisher, Bodø, Norway, 2000.