Artificial Neural Network

Kshirsagar A.P.
Assistant Professor, K.B.P.College of Engg. & Poly, Satara-415002

Rathod M.N.
Assistant Professor, K.B.P.College of Engg. & Poly, Satara-415002

ABSTRACT

A neural network is a powerful data modeling tool that is able to capture and represent complex input/output relationships. Imagine the power of the machine which has the abilities of both computers and humans. It would be the most remarkable thing ever. 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. The computing world has a lot to gain from neural networks. Their ability to learn by example makes them very flexible and powerful. 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. With the correct implementation NN can be used naturally in online learning and large dataset applications. If the 21st Century is to be the age of intelligent machines, then 'Neural Networks' will become an integral part of life.

This paper focuses on the many aspects of NN, the past, present and the future and explores what it has kept folded for us in the 'GENERATION NEXT…..'

Keywords

pattern recognition, Neuron, Human Brain, Supervised Learning, unsupervised Learning.

1. INTRODUCTION

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. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANN, like people, learns 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 neurons. This is true of ANN as well.

1.1 Neural Network Vs conventional Computers

Neural networks take a different approach to problem solving than that of conventional computers. Conventional computers use an algorithmic approach i.e. the computer follows a set of instructions in order to solve a problem. Unless the specific steps that the computer needs to follow are known the computer cannot solve the problem. That restricts the problem solving capability of conventional computers to problems that we already understand and know how to solve.

Neural networks process information in a similar way the human brain does. The network is composed of a large number of highly interconnected processing elements (neurons) working in parallel to solve a specific problem. Neural networks learn by example. They cannot be programmed to perform a specific task. The examples must be selected carefully otherwise useful time is wasted or even worse the network might be functioning incorrectly. The disadvantage is that because the network finds out how to solve the problem by itself, its operation can be unpredictable.

On the other hand, conventional computers use a cognitive approach to problem solving; the way the problem is to solved must be known and stated in small unambiguous instructions. These instructions are then converted to a high level language program and then into machine code that the computer can understand. These machines are totally predictable; if anything goes wrong is due to a software or hardware fault. Neural networks and conventional algorithmic computers are not in competition but complement each other. There are tasks are more suited to an algorithmic approach like arithmetic operations and tasks that are more suited to neural networks. Even more, a large number of tasks, require systems that use a combination of the two approaches (normally a conventional computer is used to supervise the neural network) in order to perform at maximum efficiency.

2. Analogy to Biological Neural Network

2.1 Biological Neural Network

The fundamental processing element of a neural network is a neuron. This building block of human awareness encompasses a few general capabilities. Basically, a biological neuron receives inputs from other sources, combines them in some way, performs a generally nonlinear operation on the result, and then outputs the final result. Figure 2.1 shows the relationship of these four parts.

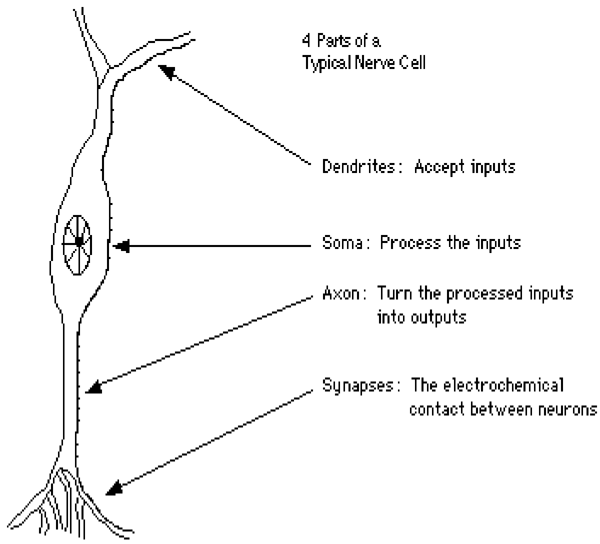

Figure 2.1 A Simple Neuron

Within humans there are many variations on this basic type of neuron, further complicating man's attempts at electrically replicating the process of thinking. Yet, all natural neurons have the same four basic components. These components are known by their biological names - dendrites, soma, axon, and synapses. Dendrites are hair-like extensions of the soma which act like input channels. These input channels receive their input through the synapses of other neurons. The soma then processes these incoming signals over time. The soma then turns that processed value into an output which is sent out to other neurons through the axon and the synapses. Recent experimental data has provided further evidence that biological neurons are structurally more complex than the simplistic explanation above. They are significantly more complex than the existing artificial neurons that are built into today's artificial neural networks. As biology provides a better understanding of neurons, and as technology advances, network designers can continue to improve their systems by building upon man's understanding of the biological brain.

2.2 Artificial Neural Network

But currently, the goal of artificial neural networks is not the grandiose recreation of the brain. On the contrary, neural network researchers are seeking an understanding of nature's capabilities for which people can engineer solutions to problems that have not been solved by traditional computing. To do this, the basic unit of neural networks, the artificial neurons, simulates the four basic functions of natural neurons. Figure 2.2.2 shows a fundamental representation of an artificial neuron.

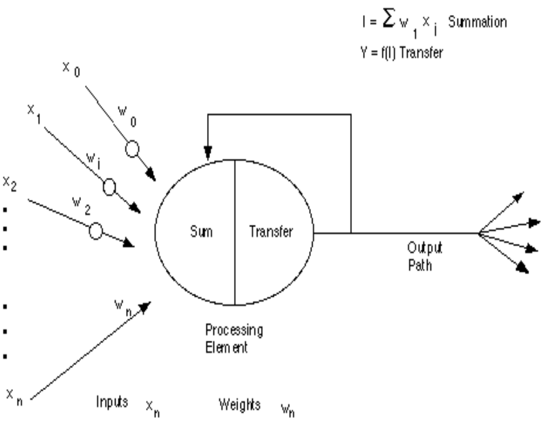

Figure 2.2 A Basic Artificial Neuron.

In Figure 2.2, various inputs to the network are represented by the mathematical symbol, x (n). Each of these inputs is multiplied by a connection weight. These weights are represented by w (n). In the simplest case, these products are simply summed, fed through a transfer function to generate a result, and then output. This process lends itself to physical implementation on a large scale in a small package. This electronic implementation is still possible with other network structures which utilize different summing functions as well as different transfer functions.

Some applications require "black and white," or binary, answers. These applications include the recognition of text, the identification of speech, and the image deciphering of scenes. These applications are required to turn real-world inputs into discrete values. These potential values are limited to some known set, like the ASCII characters or the most common 50,000 English words. Because of this limitation of output options, these applications don't always utilize networks composed of neurons that simply sum up, and thereby smooth, inputs. These networks may utilize the binary properties of ORing and ANDing of inputs. These functions, and many others, can be built into the summation and transfer functions of a network. Other networks work on problems where the resolutions are not just one of several known values. These networks need to be capable of an infinite number of responses. Applications of this type include the "intelligence" behind robotic movements. This "intelligence" processes inputs and then creates outputs which actually cause some device to move. That movement can span an infinite number of very precise motions. These networks do indeed want to smooth their inputs which, due to limitations of sensors, come in non-continuous bursts, say thirty times a second. To do that, they might accept these inputs, sum that data, and then produce an output by, for example, applying a hyperbolic tangent as a transfer functions. In this manner, output values from the network are continuous and satisfy more real world interfaces.

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

3. HUMAN AND ARTIFICIAL NEURONS - INVESTIGATING THE SIMILARITIES

3.1 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*.

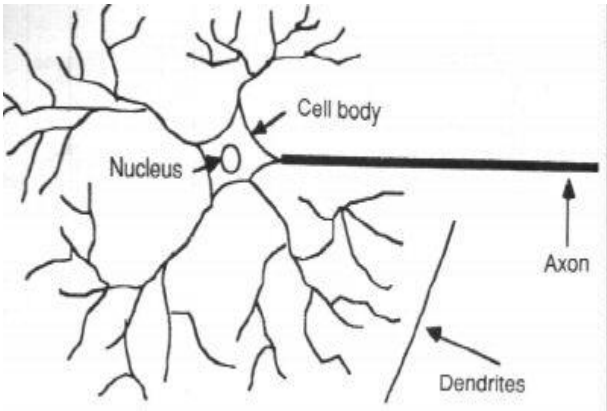

Figure 3.1 Components of a neuron

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 neurons. 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.

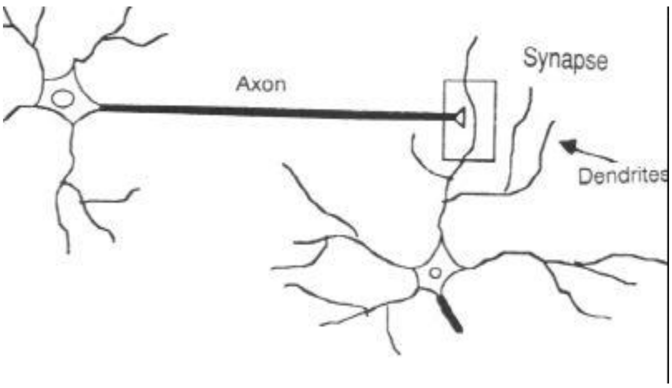

Figure 3.2 From Human Neurons to Artificial Neurons

We conduct these neural networks by first trying to deduce the essential features of neurons and their interconnections. 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.

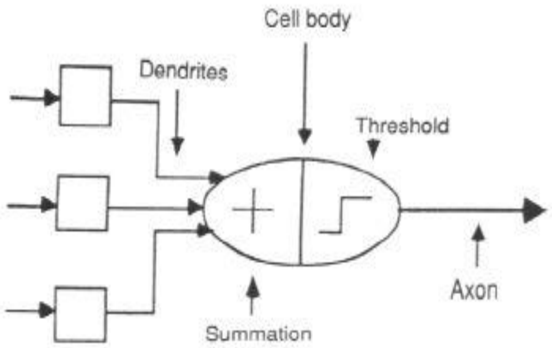

Figure 3.3 The neuron model

4. TEACHING AN ARTIFICIAL NEURAL NETWORK

4.1 Supervised Learning.

The vast majority of artificial neural network solutions have been trained with supervision. In this mode, the actual output of a neural network is compared to the desired output. Weights, which are usually randomly set to begin with, are then adjusted by the network so that the next iteration, or cycle, will produce a closer match between the desired and the actual output. The learning method tries to minimize the current errors of all processing elements. This global error reduction is created over time by continuously modifying the input weights until acceptable network accuracy is reached. With supervised learning, the artificial neural network must be trained before it becomes useful. Training consists of presenting input and output data to the network. This data is often referred to as the training set. That is, for each input set provided to the system, the corresponding desired output set is provided as well. In most applications, actual data must be used. This training phase can consume a lot of time. In prototype systems, with inadequate processing power, learning can take weeks. This training is considered complete when the neural network reaches a user defined performance level. This level signifies that the network has achieved the desired statistical accuracy as it produces the required outputs for a given sequence of inputs. When no further learning is necessary, the weights are typically frozen for the application. Some network types allow continual training, at a much slower rate, while in operation. This helps a network to adapt to gradually changing conditions.

Training sets need to be fairly large to contain all the needed information if the network is to learn the features and relationships that are important. Not only do the sets have to be large but the training sessions must include a wide variety of data. If the network is trained just one example at a time, all the weights set so meticulously for one fact could be drastically altered in learning the next fact. The previous facts could be forgotten in learning something new. As a result, the system has to learn everything together, finding the best weight settings for the total set of facts. For example, in teaching a system to recognize pixel patterns for the ten digits, if there were twenty examples of each digit, all the examples of the digit seven should not be presented at the same time.

How the input and output data is represented, or encoded, is a major component to successfully instructing a network. Artificial networks only deal with numeric input data. Therefore, the raw data must often be converted from the external environment. Additionally, it is usually necessary to scale the data, or normalize it to the network's paradigm. This pre-processing of real-world stimuli, be they cameras or sensors, into machine readable format is already common for standard computers. Many conditioning techniques which directly apply to artificial neural network implementations are readily available. It is then up to the network designer to find the best data format and matching network architecture for a given application. After a supervised network performs well on the training data, then it is important to see what it can do with data it has not seen before. If a system does not give reasonable outputs for this test set, the training period is not over. Indeed, this testing is critical to insure that the network has not simply memorized a given set of data but has learned the general patterns involved within an application.

4.2 Unsupervised Learning

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

This promising field of unsupervised learning is sometimes called self-supervised learning. These networks use no external influences to adjust their weights. Instead, they internally monitor their performance. These networks look for regularities or trends in the input signals, and makes adaptations according to the function of the network. Even without being told whether it's right or wrong, the network still must have some information about how to organize itself.

This information is built into the network topology and learning rules.

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

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

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

5. TYPES OF NEURAL NETWORKS:-

5.1 Feed forward Neural Network:

The feed forward neural network was the first and simplest type of artificial neural network. In this network, the information moves in only one direction, forward, from the input nodes, through the hidden nodes and to the output nodes. There are no cycles or loops in the network.

5.2 Radial basis function (RBF) Network:

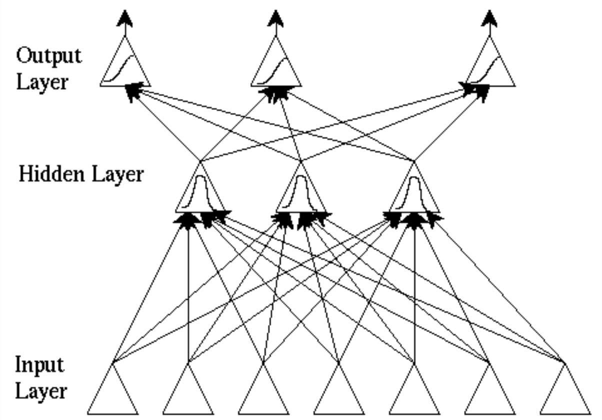

Figure 5.2 RBF Network

Radial Basis Functions are powerful techniques for interpolation in multidimensional space. A RBF is a function which has built into a distance criterion with respect to a centre. Radial basis functions have been applied in the area of neural networks where they may be used as a replacement for the sigmoidal hidden layer transfer characteristic in multi-layer perceptions. RBF networks have two layers of processing: In the first, input is mapped onto each RBF in the 'hidden' layer. In regression problems the output layer is then a linear combination of hidden layer values representing mean predicted output. The interpretation of this output layer value is the same as a regression model in statistics. RBF networks have the disadvantage of requiring good coverage of the input space by radial basis functions.

5.3 Kohonen self-organizing Network:

The *self-organizing map* (SOM) invented by *Teuvo Kohonen* uses a form of *unsupervised learning*. A set of artificial neurons learn to map points in an input space to coordinates in an output space. The input space can have different dimensions and topology from the output space and the SOM will attempt to preserve these.

5.4 Recurrent Network:

Recurrent neural networks are models with bi-directional data flow. While a feed forward network propagates data linearly from input to output, RNs also propagate data from later processing stages to earlier stages

5.2 RBF for face recognition

The salient features of RBF neural networks are as

follows.

- They are universal approximators [1].
- They possess the best approximation property [3].
- Their learning speed is fast because of locally tuned neurons [4].
- They have more compact topology than other neural networks [5].Normally, RBF neural networks are widely used for function approximation and pattern recognition wherein the pattern dimensioning these applications is usually small.

As pointed out by Moody and Darken [4], "RBF neural networks are best suited for learning to approximate continuous or piecewise continuous, real-valued mapping where the input dimension is sufficiently small."

When RBF neural networks are implemented in face recognition, such systems possess the following characteristics:

• High dimension. For example, a 128 128 image will have 10 30 features.

• Small sample sets. The sample patterns are very few for each class, say, only one–ten images per person so that (is the number of training patterns, is the number of features), which is more severe than the case shown in [10].

Therefore, face recognition is substantially different from classical pattern recognition problem, for instance, character recognition [9], in which there are a limited number of classes with a large number of training patterns in each class.

6. APPLICATIONS

Clustering

A Clustering algorithm explores the similarity between patterns and places similar patterns in a cluster. Best known applications include data compression and data mining.

6.1 *6.2Classification/Pattern recognition:*

The task of pattern recognition is to assign an input pattern (like handwritten symbol) to one of many classes. This category includes algorithmic implementations such as associative memory.

6.2 Function approximation:

The tasks of function approximation is to find an estimate of the unknown function f () subject to noise. Various engineering and scientific disciplines require function approximation.

6.3 Prediction/Dynamical System

The task is to forecast some future values of a time-sequenced data. Prediction has a significant impact on decision support systems. Prediction differs from Function approximation by considering time factor. Neural networks in medicine-: In the Medical Field, the research is mostly on modeling parts of the human body and recognizing diseases from various scans (e.g. cardiograms, CAT scans, ultrasonic scans, etc.).

6.4 Facial Expressions:

Neural networks could be use for learning of each variation in the face expressions for animated sequences.

7. ACKNOWLEDGEMENT-:

Our thanks to the Prof. Patil G.A. &Prof Manwade K.B. for their valuable guidance.

8. REFERENCES

[1] J. Park and J. Wsandberg, "Universal approximation using radial basis functions network," Neural Compute., vol. 3, pp. 246–257, 491.

[2] Y. Moses, Y. Adini, and S. Ullman, "Face recognition: The problem of compensating for changes in illumination direction," in Proc. EuroP.Conf. Compute. Vision, vol. A, 4937, pp. 286–296.

[3] F. Girosi and T. Poggio, "Networks and the best approximation property," Biol. Cybern., vol. 19, pp. 109–100, 490.

[4] J. Moody and C. J. Darken, "Fast learning in network of locally-tuned processing units," Neural Compute., vol. 1, pp. 237–2937, 489.

[5] S. Lee and R. M. Kil, "A Gaussian potential function network with hierarchically self-organizing learning," Neural Networks, vol. 37, pp. 10–620, 491.

[6] Goedel, K. (1931) "On Formally Undecidable Propositions of Principia Mathematica and Related Systems I" in Davis, M. (ed) (1965) The Undecidable. Raven Press

[7] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces versusfisherfaces: Recognition using class specific linear projection," IEEE Trans. Pattern Anal. Machine Intell, vol. 4, pp. 271–31, 497.

[8] D. L. Swets and J. Weng, "Using discriminant eigenfeatures for image retrieval," IEEE Trans. Pattern Anal. Machine Intell., vol. 3, pp.374–380, 496.

[9] H. H. Song and S. W. Lee, "A self-organizing neural tree for large-set pattern classification," IEEE Trans. Neural Networks, vol. 9, pp. 209–310, Mar. 498.

[10] J. L. Yuan and T. L. Fine, "Neural-Network design for small training sets of high dimension," IEEE Trans. Neural Networks, vol. 9, pp. 266–236,Jan. 498.