Neural networks-an introduction

Conference Paper · February 1991

Source: IEEE Xplore

John Mark Bishop
Goldsmiths, University of London

R.J. Mitchell
University of Reading

NEURAL NETWORKS - AN INTRODUCTION

J.M.Bishop and R.J.Mitchell

Abstract

The last few years have witnessed the reemergence of the field of neural networks and there is considerable work in this area all over the world. It is predicted that neural network technology will revolutionise many aspects of computing. In this paper, neural networks are introduced, a brief history of the subject is given, some of the various network techniques are described and suitable application areas are surveyed.

Neural Networks

The field of neural networks, also known as connectionism, parallel distributed processing, connection science, or neural computing, is a mode of computing which seeks to include the style of computing used in the brain. In their book *An Introduction to Neural Computing*[1], Aleksander and Morton have the following definition:

Neural computing is the study of networks of adaptable nodes which, through a process of learning from task examples, store experiential knowledge and make it available for use.

A neural network is a processor of information consisting of simple processing elements connected together. Each processing element is a very simple model of a neuron in the brain: hence the term neural networks. Thus a neural network could be described as mankind's attempt to create an artificial brain. However, present neural networks try to mimic the way the brain does things in order to harness its ability to infer and induce from incomplete or confusing information.

What makes these networks powerful is their potential for performance improvement over time as they acquire more knowledge about a problem, and their ability to handle fuzzy real world data. That is, a network 'taught' certain data patterns, is able to recognise both these patterns and those which are similar: the network is able to generalise. Also, neural networks are inherently parallel in their operation, and so have the capacity to operate much faster than conventional computers.

Work in neural networks began in the 1940s and flourished under McCulloch, Pitts, Hebb, Widrow, Minsky and Rosenblatt until 1969. Unfortunately, the claims by Rosenblatt as to what his machine could do were somewhat exaggerated, and in 1969 Minsky and Papert wrote a book, *Perceptrons*[2], which showed that there were various problems with these neural networks and so the subject went out of fashion until the mid 1980s. During this time, however, various people were still working in the field, including Aleksander, Kohonen, Hopfield, and the PDP group including Rumelhart, McClelland and Hinton. Neural networks have became fashionable again because it has been realised that there are limitations as to what Expert Systems can do, and the success of Hopfield and the PDP group. In 1986 Rumelhart and McClelland published *Parallel Distributed Processing*[3] in which it is shown how the objections of Minsky and Papert could be overcome, and work in the field has flourished.

A neural network consists of simple processing elements connected together. There are various ways in which these elements can be connected, in single or multiple layers, fully or partially connected, etc., and there are various forms of processing elements. However, many factors are common. These will be considered first, then some of the different models and topologies will be described briefly.

A neural network is not programmed to complete a given task, rather it adapts and acquires knowledge over time in order to complete the task. One stage of operation, therefore, requires the network to learn, and this can be supervised or unsupervised. Once the network has learned to perform a task, it can then be set to undertake that task. For example, a network which is to be used to recognise objects would first be taught what those objects looked like, and then would be put in the mode whereby it reported whether the object it was being shown was one it had been taught.

J.M.Bishop and R.J.Mitchell are with the Neural Network Research Group, Department of Cybernetics, School of Engineering and Information Sciences, University of Reading, Reading.

There are very many forms of neural network of which the main types are the 'Perceptron' types, Hopfield nets, Boltzmann machines, Weightless or n-tuple nets, Kohonen nets, the neocognitron and ART classifiers. The two types which are used the most, and which are more applicable in the area covered by the colloquium, are 'Perceptron' and Kohonen nets, and these are described next.

Perceptron networks

In these networks the main element is an extension of the McCulloch and Pitts neuron: this is shown in figure 1. The output of the element is some function of the weighted sum of all the inputs. Sometimes the output value is used directly, or the output is processed by, say, a threshold or the sigmoid function (as shown in figure 2). In some networks it is important to know if the element has 'fired' (in the same way that a neuron in the brain produces an active output), and this occurs if the weighted sum exceeds a given threshold. For other networks, the actual value of the output is required.

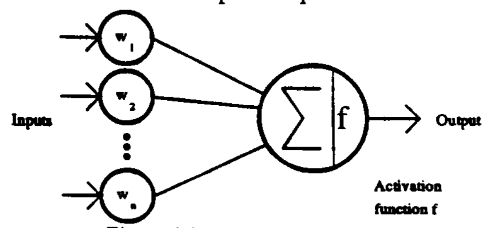

Figure 1 Basic model of Neuron

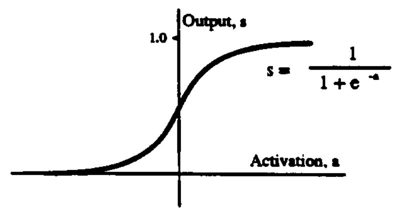

Figure 2 The Sigmoid function

Learning is the process of deciding what the values of the weights should be. This is often achieved by comparing the output of the element with what it should be (this is a form of supervised learning) and adjusting the weights appropriately. Normally the Widrow-Hoff *Delta* rule is used to adjust weights.

A single layer network of such neurons can perform simple functions. However, one of the criticisms raised by Minsky and Papert was that such a network could not solve a non linearly separable problem: that is one where one straight line cannot separate opposing classes. However, a multiple layer network can solve such problems.

In learning mode, the network is repeatedly presented the data in the training set, both the required input and output for each item in the set, and the weights adjusted accordingly using the generalised delta rule, by propogating errors between the actual output and the desired output back through the network. This can take time as the whole data set must be presented many hundred times. This is a severe disadvantage of the method. However, a great many workers in the field use back propogation learning on multi layer perceptrons.

Kohonen networks

Teuvo Kohonen postulated a different form of network with a different learning strategy. The learning of these networks is unsupervised, which in some ways is more like the form in which humans learn. In Kohonen networks each node has a number of values associated with it, one value for each input to the network, and each input is connected to each node. In use, the values associated with each node are compared with the input, and the node which most closely resembles the input is the one which fires. The system must be taught patterns to fire, and also be able to recognise similar patterns. This is achieved in the following way.

In learn mode, the node which most resembles the input is determined. Then, its weights are adjusted so as to reduce the difference between the weights and the inputs. This allows the node to resemble the input more closely. So that similar patterns are also remembered, those nodes which are connected to this node, that is those which are adjacent to it, also have their weights adjusted, but the amount by which they are adjusted is determined by the distance of each node from that which fired. If the main node is changed by an amount δ, the other nodes are changed by $M*\delta$, where M is determined by the 'mexican hat' function (shown in figure 3).

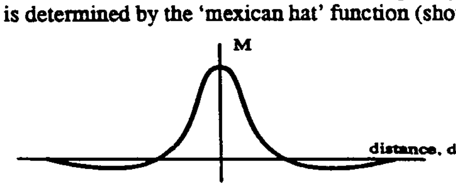

Figure 3 'Mexican Hat'

Therefore the system should be able to remember the taught input patterns, and similar patterns, that is the network can generalise. Kohonen has used these networks successfully in the field of speech recognition and many other workers use Kohonen networks.

Neural Network Applications

The various application areas of neural networks are determined by their properties. These include their ability to learn, generalisation, their distributed memory so the failure of one node does not necessarily lead to the failure of the whole system, and their inherent parallelism so they can be fast.

Given these properties, it is possible to define the following characteristics of problems which are suitable for implementation using neural networks. The rules used to solve the problem may be unknown or difficult to formalise, but suitable examples as defined by inputs and corresponding outputs are available. The problem uses noisy data. The problem may evolve from some initial condition. The problem may need high speed processing.

These characteristics suggest possible application areas for networks: pattern recognition, signal processing, speech processing, robotics, control, forecasting and modelling and decision making aids. The recent conference on neural networks in Paris 1990 (INNC-90) was divided into sections similar to the above areas, and a survey of the papers presented there illustrates typical applications of networks[5].

One example presented at Paris illustrates how networks can be used in a control application: this is Widrow's Truck Backer[6], where the system learnt to reverse a truck with a trailer. This used two 'perceptron' type networks: one learnt to model the truck and trailer, by being presented various paths the truck and trailer followed; and the other learnt to control the truck, by adjusting its weights according to the distance between the actual and desired position of the trailer when it hit the wall.

Another relevant example is recent work in the Department of Cybernetics to find the concentrations of various dyes needed to produce a desired colour[7]. Here a neural network is used to effectively build a model relating dyes to colours. This is achieved using a multi layer perceptron network.

Networks can also be used with expert systems. An expert system processes data according to specific rules, but requires the interrogation of an expert in order to generate the rules. This can be difficult as the expert often cannot define the rules he uses, but can describe what he does by example. Thus neural networks can be used to generate those rules from the examples[8].

In conclusion, neural networks are powerful devices which have significant advantages over conventional computers. Already these networks are used in a great variety of applications, and there is great scope for them in the future as useful devices in themselves or as components in more complex systems.

References

1. Aleksander, I and Morton, H. "An Introduction to Neural Computing" Chapman and Hall 1990

2. Minsky, M and Papert, S. "Perceptrons" MIT Press 1969

3. Rumelhart, D.E. and McClelland, J.L. "Parallel Distributed Processing MIT Press 1986, Vols 1 and 2

4. Kohonen, T. "Self Organisation and Associative Memory SpringerVerlag 1984

5. Proceedings of INNC-90 Paris, 1990.

6. Nguyen, D and Widrow, B. "The Truck Backer-Upper", Proceedings of INNC-90, pp399-407, July 1990.

7. Bishop J.M., Bushnell M.J. and Westland S. "Computer Recipe Prediction using Neural Networks" Expert Systems 90, pp70-78, Sept 1990.

8. Kiernan, L. and Warwick, K. "Learning systems for fault diagnosis in power networks". Proc 25. Universities Power Engineering Conference, Aberdeen, Sept 1990.