

Artificial Neural Networks

A Brief Introduction

Jitendra R Raol and Sunilkumar S Mankame

Artificial neural networks are ‘biologically’ inspired networks. They have the ability to learn from empirical data/information. They find use in computer science and control engineering fields.

In recent years artificial neural networks (ANNs) have fascinated scientists and engineers all over the world. They have the ability to learn and recall - the main functions of the (human) brain. A major reason for this fascination is that ANNs are ‘biologically’ inspired. They have the apparent ability to imitate the brain’s activity to make decisions and draw conclusions when presented with complex and noisy information. However there are vast differences between biological neural networks (BNNs) of the brain and ANNs.

A thorough understanding of biologically derived NNs requires knowledge from other sciences: biology, mathematics and artificial intelligence. However to understand the basics of ANNs, a knowledge of neurobiology is not necessary. Yet, it is a good idea to understand how ANNs have been derived from real biological neural systems (see *Figures 1,2* and the accompanying boxes). The soma of the cell body receives inputs from other neurons via adaptive synaptic connections to the dendrites and when a neuron is excited, the nerve impulses from the soma are transmitted along an axon to the synapses of other neurons. The artificial neurons are called neuron cells, processing elements or nodes. They attempt to simulate the structure and function of biological (real) neurons. Artificial neural models are loosely based on biology since a complete understanding of the behaviour of real neuronal systems is lacking. The point is that only a part of the behaviour of real neurons is necessary for their information capacity. Also, it is easier to implement/simulate simplified models rather than complex ones.



J R Raol is a scientist with NAL. His research interests are parameter estimation, neural networks, fuzzy systems, genetic algorithms and their applications to aerospace problems. He writes poems in English.



Sunilkumar S Mankame is a graduate research student from Regional Engineering College, Calicut working on direct neural network based adaptive control.



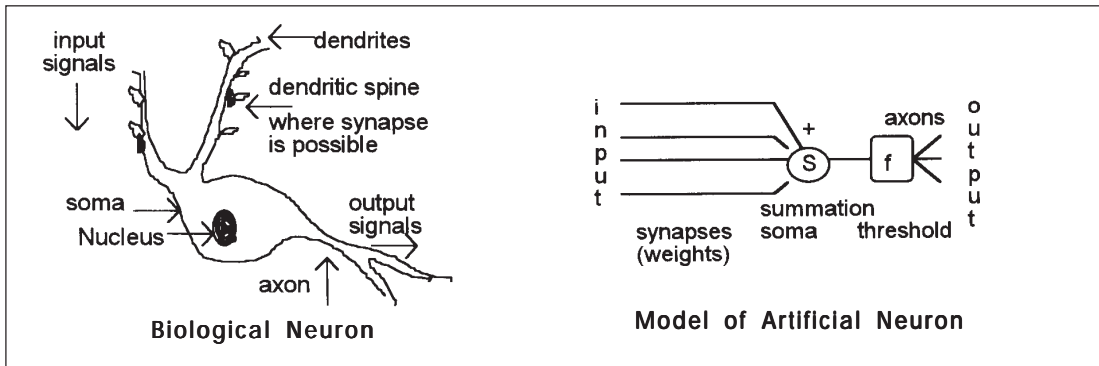


Figure 1 Biological and artificial neuron models have certain features in common. Weights in ANN model/represent synapses of BNN.

Biological Neuron System

- 1 Dendrites input branching tree of fibres - connect to a set of other neurons-receptive surfaces for input signals
- 1 Soma cell body - all the logical functions of the neurons are realised here
- 1 Synapse specialized contacts on a neuron - interfaces some axons to the spines of the input dendrites - can increase/dampen the neuron excitation
- 1 Axon nerve fibre - final output channel - signals converted into nerve pulses (spikes) to target cells

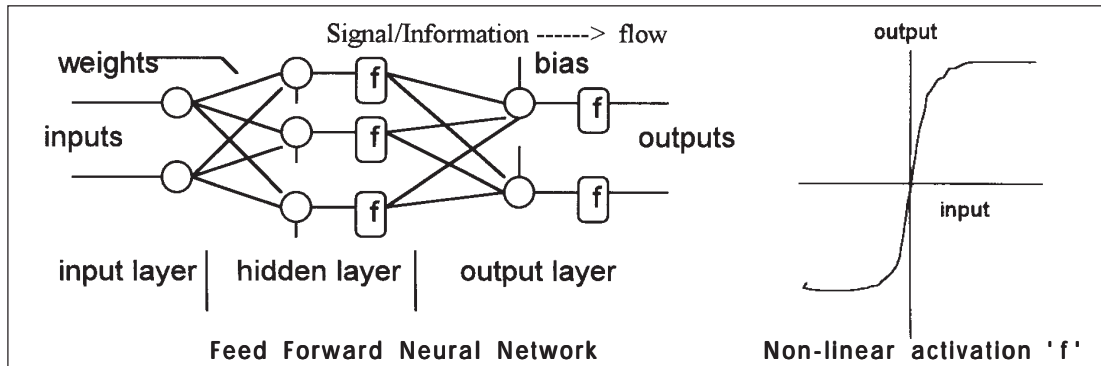
Artificial Neuron System

- 1 Input layer the layer of nodes for data entering an ANN
- 1 Hidden layer the layer between input and output layers
- 1 Output layer the layer of nodes that produce the network's output responses
- 1 Weights strength or the (gain) value of the connection between nodes

Artificial neural networks have the apparent ability to imitate the brain's activity to make decisions and draw conclusions when presented with complex and noisy information.

The first model of an elementary neuron was outlined by McCulloch and Pitts in 1943. Their model included the elements needed to perform necessary computations, but they could not realise the model using the bulky vacuum tubes of that era. McCulloch and Pitts must nevertheless be regarded as the pioneers in the field of





neural networks.

ANNs are designed to realise very specific computational tasks/problems. They are highly interconnected, parallel computational structures with many relatively simple individual processing elements (*Figure 2*). A biological neuron either excites or inhibits all neurons to which it is connected, whereas in ANNs either excitatory or inhibitory neural connections are possible. There are many kinds of neurons in BNNs, whereas in ANNs only certain types are used. It is possible to implement and study various ANN architectures using simulations on a personal computer (PC).

Types of ANNs

There are mainly two types of ANNs: feed forward neural networks (FFNNs) and recurrent neural networks (RNNs). In FFNN there are no feedback loops. The flow of signals/information is only in the forward direction. The behaviour of FFNN does not depend on past input. The network responds only to its present input. In RNN there are feedback loops (essentially FFNN with output fed back to input). Different types of neural network architectures are briefly described next.

¹ *Single-layer feed forward networks*: It has only one layer of computational nodes. It is a feed forward network since it does not have any feedback.

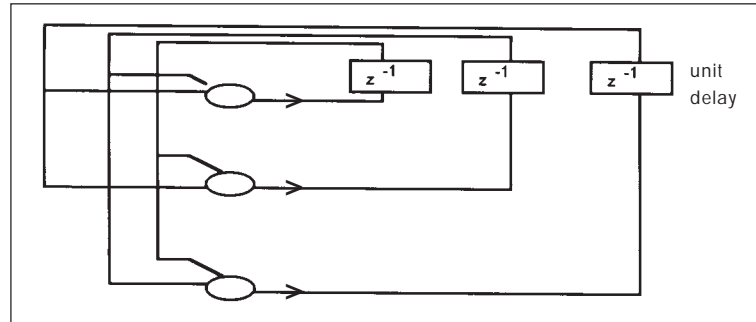
Figure 2 A typical artificial neural network (feed forward) does not usually have feedback - flow of signal/information is only in the forward direction. A multi-layer feed forward neural network (FFNN) is shown here. The non-linear characteristics of ANNs are due to the non-linear activation function f . This is useful for accurate modelling of non-linear systems..

Characteristics of non-linear systems

In non-linear systems, the output variables do not depend on the input variables in a linear manner. The dynamic characteristics of the system itself would depend on either one or more of the following: amplitude of the input signal, its wave form, its frequency. This is not so in the case of a linear system.



Figure 3 Recurrent neural network has feedback loops with output fed back to input.



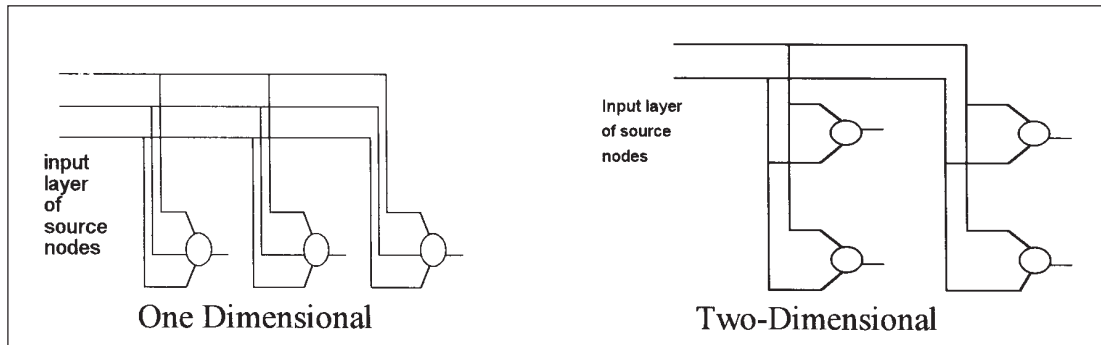
The first model of an elementary neuron was outlined by McCulloch and Pitts in 1943. The model included the elements to perform necessary computations, but they could not realise the model using the bulky vacuum tubes of that era. McCulloch and Pitts must nevertheless be regarded as the pioneers in the field of neural networks.

¹ *Multi-layer feed forward networks:* It is a feed forward network with one or more hidden layers. The source nodes in the input layer supply inputs to the neurons of the first hidden layer. The outputs of the first hidden layer neurons are applied as inputs to the neurons of the second hidden layer and so on (Figure 2). If every node in each layer of the network is connected to every other node in the adjacent forward layer, then the network is called fully connected. If however some of the links are missing, the network is said to be partially connected. Recall is instantaneous in this type of network (we will discuss this later in the section on the uses of ANNs). These networks can be used to realise complex input/output mappings.

¹ *Recurrent neural networks:* A recurrent neural network is one in which there is at least one feedback loop. There are different kinds of recurrent networks depending on the way in which the feedback is used. In a typical case it has a single layer of neurons with each neuron feeding its output signal back to the inputs of all other neurons. Other kinds of recurrent networks may have self-feedback loops and also hidden neurons (Figure 3).

¹ *Lattice networks:* A lattice network is a feed forward network with the output neurons arranged in rows and columns. It can have one-dimensional, two-dimensional or higher dimensional arrays of neurons with a corresponding set of source nodes that supply the input signals to the array. A one-dimensional network of three neurons fed from a layer of three source nodes and a two-dimensional lattice of 2-by-2 neurons fed from a layer of two source nodes are shown in Figure 4. Each source node is connected to every neuron in the lattice





network.

There are also other types of ANNs: Kohonen nets, adaptive resonance theory (ART) network, radial basis function (RBF) network, etc., which we will not consider now.

Training an ANN

A learning scheme for updating a neuron's connections (weights) was proposed by Donald Hebb in 1949. A new powerful learning law called the *Widrow-Hoff learning rule* was developed by Bernard Widrow and Marcian Hoff in 1960.

How does an ANN achieve 'what it achieves'? In general, an ANN structure is trained with known samples of data. As an example, if a particular pattern is to be recognised, then the ANN is first trained with the known pattern/information (in the form of digital signals). Then the ANN is ready to recognise a similar pattern when it is presented to the network. If an ANN is trained with a character 'H', then it must be able to recognise this character when some noisy/fuzzy H is presented to it. A Japanese optical character recognition (OCR) system has an accuracy of about 99% in recognising characters from thirteen fonts used to train the ANN. The network learns (updates its weights) from the given data while trying to minimise some cost function of the 'error' between the training set data (output) and its own output. This learning is accomplished by the popular back-propagation (BPN) algorithm, which is slow in convergence. It is a learning method in which an output error is reflected

Figure 4 Lattice networks are FFNNs with output neurons arranged in rows and columns. The one-dimensional network of three neurons shown here is fed from a layer of three source nodes. The two-dimensional lattice is fed from a layer of two source nodes.

ANNs are used for pattern recognition, image processing, signal processing/prediction, adaptive control and other related problems. 51



How does an ANN achieve 'what it achieves'? In general, an ANN structure is trained with known samples of data. For example, if a particular pattern is to be recognised, then the ANN is first trained with the known pattern/information (in the form of digital signals). Then the ANN is ready to recognise a similar pattern when it is presented to the network.

to the hidden layers (for updating the hidden layer weights). In essence, the numerical values of the weights (strengths of synapses; see the box and *Figure 2*) of the network are updated using some rule of the type:

$$\text{new weights} = \text{old (previous) weights} + \text{learning rate times gradient of the cost function (with respect to the weights).}$$

Once the ANN is trained, the 'learned information' is captured in these weights in a condensed (and yet complex) way. In general, these weights have no direct physical significance or meaning related to the process or phenomenon which is described or modelled by the ANN. However, the overall function of the ANN is relevant to a given task/application.

Although the basic processing element is simple, when several of them are connected (and also needed in many applications), the size of an ANN could be very large in terms of the number of neurons (hundreds) in a given layer. A maximum of one hidden layer is often sufficient for many tasks, to attain the required accuracy. The larger the ANN, the more time it takes to train the network. Despite the simplicity of the basic units, the mathematical theory to study and analyse the various structures and schemes based on ANNs and their (non-linear) behaviour can be very involved.

Uses of ANNs

The ANNs are used for pattern recognition, image processing, signal processing/prediction, adaptive control and other related problems. Due to the use of non-linear activation functions, the ANNs are highly suitable for very accurate mapping (modelling) of non-linear systems based on the input/output data. This non-linear activation function is also useful in reducing the adverse effect of outliers/spikes in data, and in improving the accuracy of estimates obtained by RNN (see the box).

Some of the uses of ANNs (FFNNs and RNNs) are briefly men-



tioned here:

- 1 Information storage/recall - The recall is a process of decoding the previously stored information by the network.
- 1 Pattern recognition/classification - to recognise a specific pattern from a cluster of data/ to classify sets of data/information.
- 1 Non-linear mapping between high-dimensional spaces (mathematical modelling of non-linear behaviour of systems).
- 1 Time-series prediction (like weather forecasting), modelling of non-linear aerodynamic phenomena, detection of faults/failures in systems like power plants and aircraft sensors.

Some of the specific applications (based on open literature)/possibilities are:

- 1 Assisting IC-CIM (computer integrated manufacturing of integrated circuits).
- 1 Japanese-OCR (optical character recognition).
- 1 Financial forecasting (the Neuro-Forecasting Centre in London).
- 1 Process control (Fujitsu Ltd., Kawasaki, Japan).
- 1 Analysis of medical tests.
- 1 Target tracking and recognition (multi-sensor data fusion).

In the field of artificial intelligence, we have heard of expert systems (ES). In essence an ES is a software-based system that describes the behaviour of (human) experts in some field by capturing/collecting the knowledge in the form of rules and symbols. If fuzzy or noisy data is given to an ES, it might give wrong answers. Since the ANN-based system can be trained with some fuzzy or noisy data, the combination of ES and ANN (hybrid) might be very useful to devise powerful systems called expert networks (ENs).

Concluding Remarks

By now it must be clear that in ANNs physiological or chemical processes play no role. ANNs can be called massively parallel adap-

RNNs

RNNs are specially suitable for estimation of parameters of dynamic systems in an explicit way. The RNN-based schemes for parameter estimation have been shown (by the first author) to be the generalisation of some of the conventional parameter estimation methods. These methods are useful for estimation of parameters of dynamic systems in explicit ways, e.g. mathematical modelling of aircraft dynamics. When realised in hardware, the RNN architectures are such that they are naturally adapted to obtain fast solutions to parameter estimation problems (depending on the speed of the basic processing hardware elements). RNNs are thus specially suited for arriving at non-linear state space models of dynamic systems.



The days of neural network-based (parallel) computers may not be too far off. Thus brain to bread (industrial applications) neural computers will gradually become a reality.

tive filters/circuits (MAPAFS). They are more like electrical/electronic (EE) circuits with some useful adaptive properties. In fact ANNs can be realised using EE hardware components. With very large scale integration (VLSI) technology, it might be feasible to fabricate microelectronic networks of high complexity for solving many optimisation and control problems using artificial neural networks. The days of neural network-based (parallel) computers may not be too far off. Thus brain to bread (-industrial applications) neural computers will gradually become a reality. As of now the field is fascinating, sometimes intriguing, and offers great challenges and promises to scientists and engineers.

Address for correspondence

J R Raol,
Flight Mechanics and
Control Division,
National Aerospace
Laboratories, PB 1779,
Bangalore 560017, India.

Suggested Reading

J M Zurada. *Introduction to Artificial Neural Systems.* West Publishing Company, New York. 1992.

S Haykin. *Neural Networks - A Comprehensive Foundation.* IEEE, New York. 1994.

B Kosko. *Neural Networks and Fuzzy Systems - A Dynamical Systems Approach to Machine Intelligence.* Prentice Hall, Englewood Cliffs, N.J. 1992.

R C Eberhart and R W Dobbins. *Neural Network PC Tools - A Practical Guide.* Academic Press Inc., New York. 1992.



The origin of ideas ... Ideas come when stepping onto a bus (Poincaré), attending the theatre (Wiener), walking up a mountain (Littlewood), sitting at the shore (Aleksandrov), or walking in the rain (Littlewood), but only after a long struggle of intensive work. (from *The Mathematical Intelligencer* 17(2), 1995).

