Artificial Neural Networks: - Making Machines to Learn

Jashvant R. Dave
Lecturer, NSVKMS MCA College, Visnagar
Mailing Address: 511/2, Nandanvan Society, Sector-22, Gandhinagar
Phone No: 02765-225505 (O) 079-23244551 (R) 9879117261 (M)
E-Mail Id: jashvant.dave@gmail.com

Jigneshkumar M. Patel
Lecturer Cum Programmer, NSVKMS MCA College, Visnagar (North-Gujarat)
Mailing Address: - 24, Marutinandan, Sahibag, Kansa Road, Visnagar-384315
Phone no :- (O) :- (02765)227345 (R):- (02765)226645, (Mo):- 9913234409
Email-Id: - jigneshpatel2004@hotmail.com

What is a Neural Network?

An Artificial Neural Network (ANN) is an information processing model that is based on the way biological nervous systems, such as the brain process information. The key element of this model is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in group to solve specific problems. ANN learns by example as people can. We can configure ANN for a specific application, such as pattern recognition through a learning process.

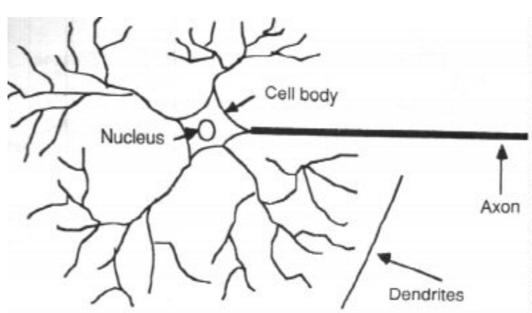
The Root for Artificial Neuron – The Human Brain

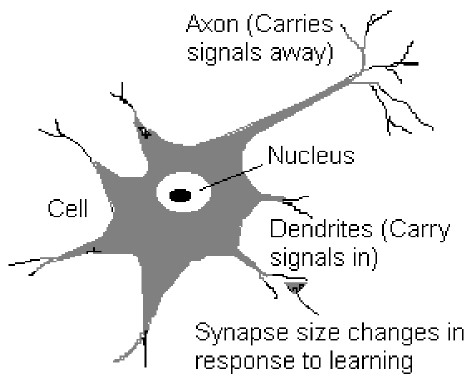
Components of Biological Neuron:

Dendrites: - A typical neuron collects signals from others through a host of fine structures that structure is called as dendrites.

Axon: - The neuron sends out spikes of electrical activity through a long, thin stand that stand is known as an axon, which splits into thousands of branches.

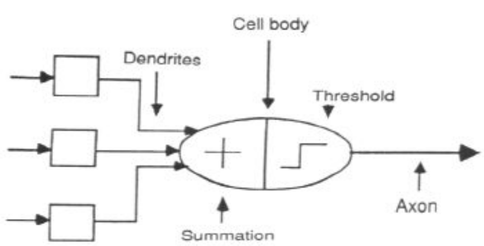
Synapse: - At the end of each branch of axon, a structure called a synapse is situated. This converts the activity from the axon into electrical effects that inhibit or excite activity in the connected neurons. When a neuron receives excitatory input that is sufficiently large compared with its inhibitory 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.





Terminology of Artificial Neuron

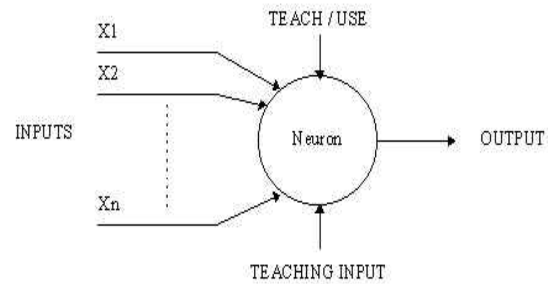
We are just going to develop an artificial neuron based on the concept of biological neuron. As like biological neuron our artificial neuron contains dendrite which takes the input from outside. In artificial neuron we will input value X as dendrites. Mathematical functions and threshold (Range of value) value works as cell body and answer of this simulation is axon.



The Basic Neuron Model

A simple neuron

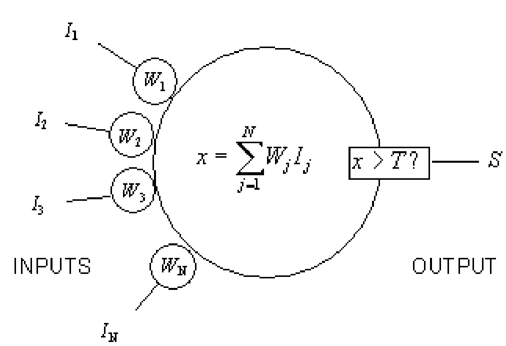
An artificial neuron is a device with many inputs and one output. The neuron has two modes of operation; the training mode and the using mode. In the training mode, the neuron can be trained to fire (or not), for particular input patterns. In the using mode, when a taught input pattern is detected at 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.

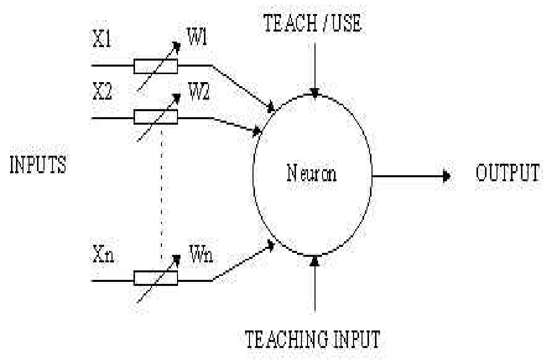


A simple neuron

A more complicated neuron

The previous neuron's functionality is very simple and we can not get accurate result using that. So, a more sophisticated neuron is mentioned here. 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 (noted as T) the neuron fires. In any other case the neuron does not fire.





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

$$X_1W_1 + X_2W_2 + X_3W_3 + ... > T$$

The addition of input weights and of the threshold makes this neuron a very flexible and powerful one. The 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.

Architecture of Neural Networks

Construction of a Neural Network involves the following tasks:

- Determine the Network Properties
- Determine the Node Properties
- Determine the System Dynamics

Network Properties:

The topology of a neural network refers to its framework as well as its interconnection scheme. The framework is often specified by the number of layers and number of nodes per layer. The types of layer include:

Input Layer: - The nodes in it are called input units; the activity of the input layer represents the raw information that is fed into the network for processing.

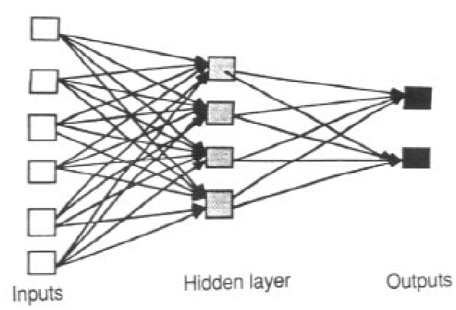
Hidden Layer: - The nodes in it are called hidden units, which are not directly observable and hence hidden. They provide nonlinearities for the network.

Output Layer: - The nodes in it are called output units, which encode possible concepts to be assigned to the instance under consideration.

This simple type of network is interesting because the hidden units are free to construct their own representations of the input. The weights between the input and hidden units determine when each hidden unit is active, and so by modifying these weights, a hidden unit can choose what it represents.

Feed-forward networks

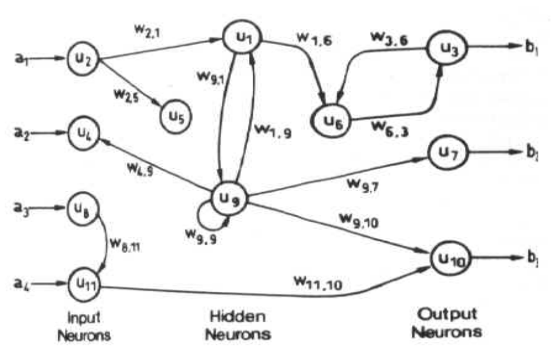
Feed-forward ANN allow signals to travel one way only from input to output. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. Feed-forward ANN tends to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition. This type of organization is also referred to as bottom-up or top-down.



An Example of Feed Forward Network

Feedback networks

Feedback networks can have signals traveling in both directions by introducing loops in the network. Feedback networks are very powerful and can get extremely complicated. Feedback networks are dynamic their 'state' is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Feedback architectures are also referred to as interactive or recurrent, although the latter term is often used to denote feedback connections in single-layer organizations.



An example of a feedback network

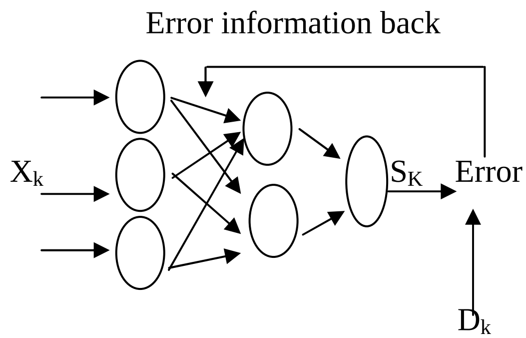
The Learning Process

Learning would be enabling the neural network to improve their performance automatically over time. So learning is very important part of the neural network.

All learning methods used for adaptive neural networks can be classified into two major categories:

Supervised learning

The supervised learning incorporates an external teacher, so that each output unit is told what its desired response to input signals ought to be. During the learning process global information may be required. Paradigms of supervised learning include error-correction learning & reinforcement learning. An important issue concerning supervised learning is the problem of error convergence, i.e. the minimization of error between the desired and computed unit values. The aim is to determine a set of weights which minimizes the error. In the supervised learning output (S_k) will be compared with the desire output (D_k) and the generated error will be given back to the host neuron. Rules take the error and set weighting factors according to that. At every iteration learning output will move towards the desired output and error will be less.



Unsupervised learning

Unsupervised learning uses no external teacher and is based upon only local information. It is also referred to as self-organization, in the sense that it self-organizes data presented to the network and detects their emergent collective properties.

We say that a neural network learns off-line if the learning phase and the operation phase are distinct. A neural network learns on-line if it learns and operates at the same time. Usually, supervised learning is performed off-line, whereas unsupervised learning is performed on-line.

Transfer Function

The behavior of an ANN (Artificial Neural Network) depends on both the weights and the input-output function (transfer function) that is specified for the units. This function typically falls into one of three categories:

- Linear (or ramp)
- Threshold
- Sigmoid

For **linear units**, the output activity is proportional to the total weighted output.

For **threshold units**, the output are set at one of two levels, depending on whether the total input is greater than or less than some threshold value.

For **sigmoid units**, the output varies continuously but not linearly as the input changes. Sigmoid units bear a greater resemblance to real neurons than do linear or threshold units, but all three must be considered rough approximations.

To make a neural network that performs some specific task, we must choose how the units are connected to one another, and we must set the weights on the connections appropriately. The connections determine whether it is possible for one unit to influence another. The weights specify the strength of the influence.

We can teach a three-layer network to perform a particular task by using the following procedure:

1. We present the network with training examples, which consist of a pattern of activities for the input units together with the desired pattern of activities for the output units.
2. We determine how closely the actual output of the network matches the desired output.
3. We change the weight of each connection so that the network produces a better approximation of the desired output.

The Back-Propagation Algorithm

In order to train a neural network to perform some task, we must adjust the weights of each unit in such a way that the error between the desired output and the actual output is reduced. This process requires that the neural network compute the error derivative of the weights (**EW**). In other words, it must calculate how the error changes as each weight is increased or decreased slightly. The back propagation algorithm is the most widely used method for determining the **EW**.

The back-propagation algorithm is easiest to understand if all the units in the network are linear. The algorithm computes each **EW** by first computing, the rate at which the error changes as the activity level of a unit (**EA**) is changed. For output units, the **EA** is simply the difference between the actual and the desired output. To compute the **EA** for a hidden unit in the layer just before the output layer, we first identify all the weights between that hidden unit and the output units to which it is connected. We then multiply those weights by the **EA**s of those output units and add the products. This sum equals the **EA** for the chosen hidden unit. After calculating all the **EA**s in the hidden layer just before the output layer, we can compute the **EA**s for other layers in the same fashion, moving from layer to layer in a direction opposite to the way activities propagate through the network. This is what gives back propagation its name. Once the **EA** has been computed for a unit, it is straight forward to compute the **EW** for each incoming connection of the unit. The **EW** is the product of the EA and the activity through the incoming connection.

Note that for non-linear units, the back-propagation algorithm includes an extra step. Before back-propagating, the **EA** must be converted into the **EI**, the rate at which the error changes as the total input received by a unit is changed.

Applications of neural networks

Neural Networks in Practice

Given this description of neural networks and how they work, what real world applications are they suited for? Neural networks have broad applicability to real world business problems. In fact, they have already been successfully applied in many industries.

Since neural networks are best at identifying patterns or trends in data, they are well suited for prediction or forecasting needs including:

- Sales forecasting
- Industrial process control
- Customer research
- Data validation
- Risk management
- Target marketing
- Business
    - Credit Evaluation
    - Marketing
- Medical Diagnosis
    - Modeling and Diagnosing Cardiovascular System
    - Electronic Noses
    - Instance Physician

But to give you some more specific examples: ANN are also used in the following specific paradigms: recognition of speakers in communications, diagnosis of hepatitis, recovery of telecommunications from faulty software, interpretation of multi meaning Chinese words, undersea mine detection, texture analysis, three-dimensional object recognition, hand-written word recognition, and facial recognition.

Conclusion

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 is 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, AI, fuzzy logic and related subjects.

Reference:

1. Neural Networks by Satish Kumar
2. Neural Networks in Computer Intelligence by LiMin Fu
3. An introduction to neural computing. Aleksander, I. and Morton, H. 2nd edition
4. Components of Neural Network http://en.wikipedia.org/wiki/Neural_network
5. Architecture of Neural Network http://www.cs.stir.ac.uk/~lss/NNIntro/InvSlides.html
6. Neural Networks by Eric Davalo and Patrick Naim