e-PGPathshala
Subject : Computer Science
Paper: Data Analytics
Module No 17: CS/DA/17 - Data Analysis Foundations – Neural Networks
Quadrant 1 – e-text

1.1 Introduction

Neural networks process information in a similar way the human brain does. Neural networks take a different approach to problem solving than that of conventional computers. Artificial neural networks (ANNs) or connectionist systems are a computational model used in machine learning, computer science and other research disciplines, which is based on a large collection of connected simple units called artificial neurons, loosely analogous to axons in a biological brain. Connections between neurons carry an activation signal of varying strength. If the combined incoming signals are strong enough, the neuron becomes activated and the signal travels to other neurons connected to it. Such systems can be trained from examples, rather than explicitly programmed, and excel in areas where the solution or feature detection is difficult to express in a traditional computer program. Like other machine learning methods, neural networks have been used to solve a wide variety of tasks, like computer vision and speech recognition, that are difficult to solve using ordinary rule-based programming. This chapter gives an overview on neural networks.

1.2 Learning Outcomes

- Learn the basics of Neural Network
- Know applications of Neural Network

1.3 Neural network – The Big Picture

Neural network is a computer modeling approach to computation that is loosely based upon the architecture of the brain. Many different models are available, but in general they have the following components:

- Multiple, individual “nodes” or “units” that operate at the same time (in parallel)
- A network that connects the nodes together
- Information is stored in a distributed fashion among the links that connect the nodes
- Learning can occur with gradual changes in connection strength

The simplest definition of a neural network, more properly referred to as an 'artificial' neural network (ANN), is provided by the inventor of one of the first neuro computers:

> "...a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs.”
>
> - Dr. Robert Hecht-Nielsen

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 (neurones) working in unison to solve specific problems. ANNs, like people, learn 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 neurones. This is true of ANNs as well.

1.3.1 Neural network history

In 1943, neurophysiologist Warren McCulloch and mathematician Walter Pitts wrote a paper on how neurons might work. In order to describe how neurons in the brain might work, they modeled a simple neural network using electrical circuits.

In 1949, Donald Hebb wrote *The Organization of Behavior*, a work which pointed out the fact that neural pathways are strengthened each time they are used, a concept fundamentally essential to the ways in which humans learn.

As computers became more advanced in the 1950's, it was finally possible to simulate a hypothetical neural network. The first step towards this was made by Nathanial Rochester from the IBM research laboratories. Unfortunately for him, the first attempt to do so failed.

In 1959, Bernard Widrow and Marcian Hoff of Stanford developed models called "ADALINE" and "MADALINE." ADALINE was developed to recognize binary patterns so that if it was reading streaming bits from a phone line, it could predict the next bit. MADALINE was the first neural network applied to a real world problem, using an adaptive filter that eliminates echoes on phone lines. While the system is as ancient as air traffic control systems, like air traffic control systems, it is still in commercial use.

In 1962, Widrow & Hoff developed a learning procedure that examines the value before the weight adjusts it (i.e. 0 or 1) according to the rule: Weight Change = (Pre-Weight line value) * (Error / (Number of Inputs)).

In 1972, Kohonen and Anderson developed a similar network independently of one another. They both used matrix mathematics to describe their ideas but did not realize that what they were doing was creating an array of analog ADALINE circuits. The neurons are supposed to activate a set of outputs instead of just one.

The first multilayered network was developed in 1975, an unsupervised network.
In 1982, interest in the field was renewed. John Hopfield of Caltech presented a paper to the National Academy of Sciences. His approach was to create more useful machines by using bidirectional lines. Previously, the connections between neurons was only one way.

That same year, Reilly and Cooper used a "Hybrid network" with multiple layers, each layer using a different problem-solving strategy.

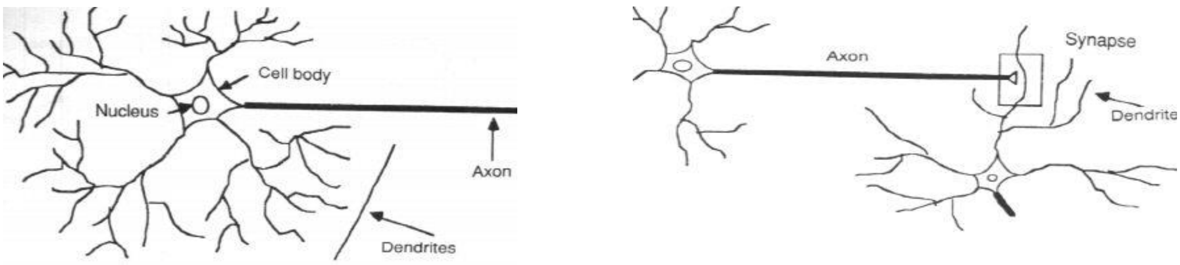
Also in 1982, there was a joint US-Japan conference on Cooperative/Competitive Neural Networks. Japan announced a new Fifth Generation effort on neural networks, and US papers generated worry that the US could be left behind in the field. (Fifth generation computing involves artificial intelligence.)

In 1986, with multiple layered neural networks in the news, the problem was how to extend the Widrow-Hoff rule to multiple layers. Hybrid networks used just two layers, these back-propagation networks use many. The result is that back-propagation networks are "slow learners," needing possibly thousands of iterations to learn.

Now, neural networks are used in several applications. The fundamental idea behind the nature of neural networks is that if it works in nature, it must be able to work in computers. The future of neural networks, though, lies in the development of hardware.

1.3.2 Neurons in brain

Much is still unknown about how the brain trains itself to process information, so theories abound. In the human brain, a typical neuron (structure is shown in figure 1 a and b) collects signals from others through a host of fine structures called dendrites. 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 neurones. 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.



(a) (b)

Figure 1. Structure of Neuron

1.3.3. Learning in the brain

The human brain consists of special cells called neurons, which are composed of several parts, including brain fibers known as dendrites. As we learn, these brain fibers grow. The fibers connect our brain cells to one another at contact points called synapses. Brains learn by altering strength between neurons either by creating/deleting connections.

Hebbian learning is one of the oldest learning algorithms, and is based in large part on the dynamics of biological systems

- Hebb's Postulate (Hebbian Learning)
  - When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased

In neuroscience, long-term potentiation (LTP) is a persistent strengthening of synapses based on recent patterns of activity. These are patterns of synaptic activity that produce a long-lasting increase in signal transmission between two neurons.

LTP is the cellular basis for learning and memory. LTP is the long-lasting strengthening of the connection between two nerve cells in response to stimulation. It is – Discovered in many regions of the cortex.

1.3.4 Computers vs Neural Networks

Neural Networks	Digital Computers
• Inductive Reasoning. Given input and output data (training examples), we construct the rules. • Computation is collective, asynchronous, and parallel. • Memory is distributed, internalized, short term and content addressable. • Fault tolerant, redundancy, and sharing of responsibilities. • Inexact. • Dynamic connectivity. • Applicable if rules are unknown or complicated, or if data are noisy or partial.	• Deductive Reasoning. We apply known rules to input data to produce output. • Computation is centralized, synchronous, and serial. • Memory is packetted, literally stored, and location addressable. • Not fault tolerant. One transistor goes and it no longer works. • Exact. • Static connectivity. • Applicable if well defined rules with precise input data.

1.3.5 Fields related with NN

(a) Computer scientists want to find out about the properties of non-symbolic information processing with neural nets and about learning systems in general.
(b) Statisticians use neural nets as flexible, nonlinear regression and classification models.
(c) Engineers of many kinds exploit the capabilities of neural networks in many areas, such as signal processing and automatic control.
(d) Cognitive scientists view neural networks as a possible apparatus to describe models of thinking and consciousness (High-level brain function).
(e) Neuro-physiologists use neural networks to describe and explore medium-level brain function (e.g. memory, sensory system).
(f) Physicists use neural networks to model phenomena in statistical mechanics and for a lot of other tasks.
(g) Biologists use Neural Networks to interpret nucleotide sequences.
(h) Philosophers and some other people may also be interested in Neural Networks for various reasons

1.3.6 Modeling a perceptron network

Perceptron (as shown figure 2) is a single artificial neuron that computes itself its weighted input and uses a threshold activation function. It is a computer model or computerized machine devised to represent or simulate the ability of the brain to recognize and discriminate**.** The most basic form of an activation function is a simple binary function that has only two possible results.

$$f(x) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

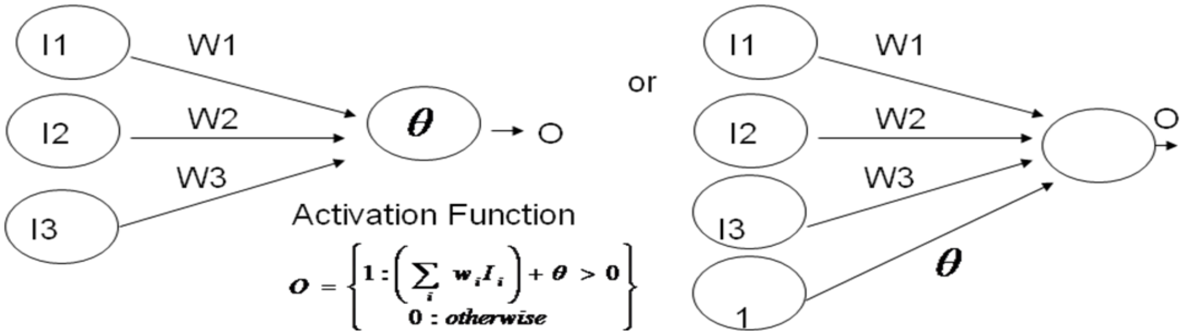


Figure 2. Perceptron

There is indeed a class of problems that a single perceptron can solve. Consider the input vector as the coordinates of a point. For a vector with n elements, this point would live in an n-dimensional space.

Further consider that we draw a number of random points on this plane, and we separate them into two sets by drawing a straight line across the paper. This line divides the points into two sets, one above and one below the line. (The two sets are then called linearly separable as shown in figure 3).

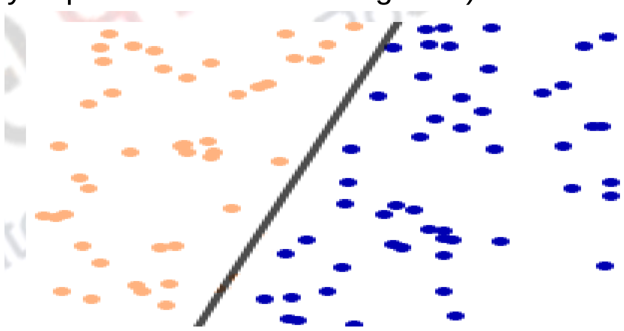


Figure 3. Linearly separable problem

A single perceptron, as bare and simple as it might appear, is able to learn where this line is, and when it finished learning, it can tell whether a given point is above or below that line.

1.3.7 Neural network advantages and drawbacks

Pros:

- prediction accuracy is generally high
- robust, works when training examples contain errors

– output may be discrete, real-valued, or a vector of several discrete or real-valued attributes
– fast evaluation of the learned target function
–

Cons:

– long training time, i.e. you need a lot of chips and a distributed run-time to train on very large datasets.
– difficult to understand the learned function (weights)
– not easy to corporate domain knowledge

1.4 Network Training

The ultimate objective of training a neural network is to obtain a set of weights that makes almost all the tuples in the training data classified correctly. The steps involved are given below:

1. Initialize weights with random values
2. Feed the input tuples into the network one by one
3. For each unit
    a. Compute the net input to the unit as a linear combination of all the inputs to the unit
    b. Compute the output value using the activation function
    c. Compute the error
    d. Update the weights and the bias

1.5 Multi-Layer Perceptron

There are many different kinds of neural networks and neural network algorithms. The most popular neural network algorithm is backpropagation, which gained repute in the 1980s.

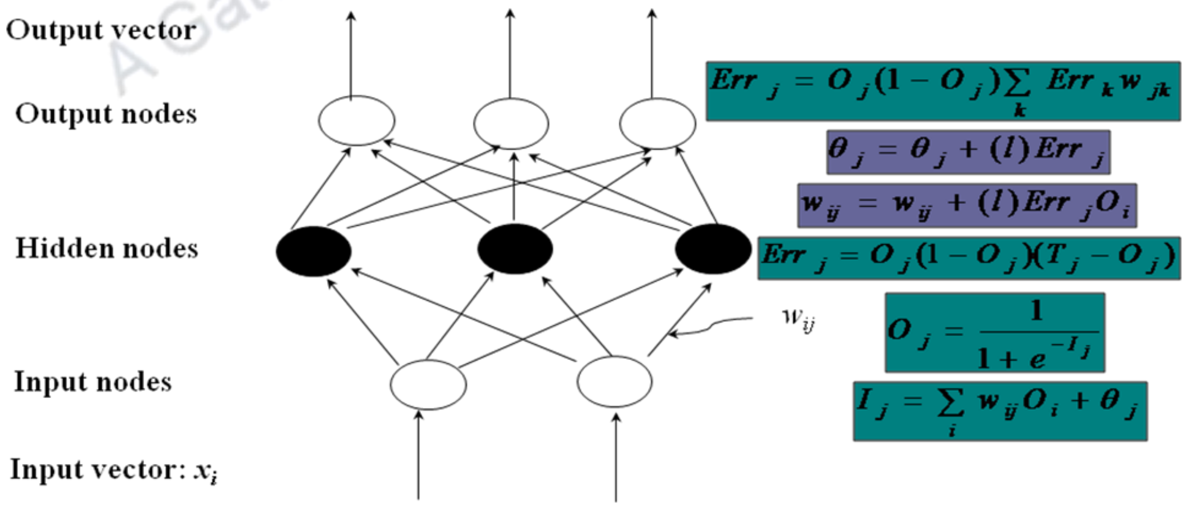


Figure 4. A multilayer feed-forward neural network

The backpropagation algorithm performs learning on a multilayer feed-forward neural network. It iteratively learns a set of weights for prediction of the class label of tuples. A multilayer feed-forward neural network consists of an input layer, one or more hidden layers, and an output layer. An example of a multilayer feed-forward network is shown in figure 4. Each layer is made up of units. The inputs to the network correspond to the attributes measured for each training tuple. The inputs are fed simultaneously into the units making up the input layer. These inputs pass through the input layer and are then weighted and fed simultaneously to a second layer of “neuronlike” units, known as a hidden layer. The outputs of the hidden layer units can be input to another hidden layer, and so on. The number of hidden layers is arbitrary, although in practice, usually only one is used. The weighted outputs of the last hidden layer are input to units making up the output layer, which emits the network's prediction for given tuples. The units in the input layer are called input units. The units in the hidden layers and output layer are sometimes referred to as neurodes, due to their symbolic biological basis, or as output units. The network shown in figure 4 is a two layered network since it contains a hidden or middle layer

Algorithm: Backpropagation. Neural network learning for classification or prediction, using the backpropagation algorithm.

Input:

- D, a data set consisting of the training tuples and their associated target values;
- l, the learning rate;
- *network*, a multilayer feed-forward network.

Output: A trained neural network.

Method:

(1) Initialize all weights and biases in *network*;
(2) while terminating condition is not satisfied {
(3) for each training tuple X in D {
(4)   // Propagate the inputs forward:
(5)   for each input layer unit j {
(6) $O_j = I_j$; // output of an input unit is its actual input value
(7)   for each hidden or output layer unit j {
(8) $I_j = \sum_i w_{ij} O_i + \theta_j$; //compute the net input of unit j with respect to the previous layer, i
(9) $O_j = \frac{1}{1+e^{-I_j}}$; } // compute the output of each unit j
(10)   // Backpropagate the errors:
(11)   for each unit j in the output layer
(12) $Err_j = O_j(1-O_j)(T_j - O_j)$; // compute the error
(13)   for each unit j in the hidden layers, from the last to the first hidden layer
(14) $Err_j = O_j(1-O_j)\sum_k Err_k w_{jk}$; // compute the error with respect to the next higher layer, k
(15)   for each weight w_{ij} in *network* {
(16) $\Delta w_{ij} = (l) Err_j O_i$; // weight increment
(17) $w_{ij} = w_{ij} + \Delta w_{ij}$; } // weight update
(18)   for each bias θ_j in *network* {
(19) $\Delta \theta_j = (l) Err_j$; // bias increment
(20) $\theta_j = \theta_j + \Delta \theta_j$; } // bias update
(21)   } }

and an output layer. The back propogation algorithm using the neural network is shown in algorithm in the previous page.

1.6. Types of Neural Networks

Neural networks can be broadly classified as: Dynamic neural network, Static neural network, Memory netork and other types. Few examples for each type os given below:

Dynamic Neural Network

- Feedforward neural network (FNN)
- Recurrent neural network (RNN)
  - Hopfield network
  - Boltzmann machine
  - Simple recurrent networks
  - Echo state network
  - Long short-term memory
  - Bi-directional RNN
  - Hierarchical RNN
  - Stochastic neural networks
- Kohonen Self-Organizing Maps
- Auto encoder
- Probabilistic neural network (PNN)
- Time delay neural network (TDNN)
- Regulatory feedback network (RFNN)

Static Neural Network

- Neocognitron
- McCulloch-Pitts cell
- Radial basis function network (RBF)
- Learning vector quantization
- Perceptron
  - Adaline model
  - Convolutional neural network (CNN)
- Modular neural networks
  - Committee of machines (COM)
  - Associative neural network (ASNN)

Memory Network

- Google / Deep Mind
- facebook / MemNN
- Holographic associative memory
- One-shot associative memory
- Neural Turing Machine
- Adaptive resonance theory
- Hierarchical temporal memory

Other types of networks

- Instantaneously trained neural networks (ITNN)
- Spiking neural network (SNN)
  - Pulse Coded Neural Networks (PCNN)
- Cascading neural networks
- Neuro-fuzzy networks
- Growing Neural Gas (GNG)
- Compositional pattern-producing networks
- Counterpropagation network
- Oscillating neural network
- Hybridization neural network
- Physical neural network
  - Optical neural network

1.7 Network Pruning and Rule Extraction

Network pruning

- Fully connected network will be hard to articulate
- *N* input nodes, *h* hidden nodes and *m* output nodes lead to *h(m+N)* weights
- Pruning: Remove some of the links without affecting classification accuracy of the network

Extracting rules from a trained network

- Discretize activation values; replace individual activation value by the cluster average maintaining the network accuracy
- Enumerate the output from the discretized activation values to find rules between activation value and output
- Find the relationship between the input and activation value
- Combine the above two to have rules relating the output to input

1.8 Applications

In **Science and medicine** domain NN is used for modeling, prediction, diagnosis, pattern recognition, e.g. effects and undesirable effects of drugs early tumor recognition. It is used for process modeling and analysis **manufacturing industries.** In **Marketing and Sales** used for analysis, classification, customer targeting, e.g. turnover prognosis for individual articles/stores.

In **Finance** NN is used for portfolio trading, investment support type of decision making. For **Banking & Insurance,** credit and policy approval, e.g. credit-scoring (Basel II), finance time series prediction, valuation of derivates, risk minimised trading strategies; client valuation e.g. risk and cost prediction for individual clients, probability of contract cancellation, fraud recognition, justice in tariffs. Also it is used in **Security** applications like bomb, iceberg, fraud detection. In **Engineering it is used for** dynamic load scheduling, pattern recognition.

	Case Studies
A) http://www.uni-obuda.hu/journal/Ogwueleka_Misra_Ogwueleka_51.pdf	
An Artificial Neural Network Model for Road Accident Prediction: A Case Study of a Developing Country An design of an Artificial Neural Network (ANN) model for the analysis and prediction of accident rates in a developing country.	
B)http://ucanalytics.com/blogs/artificial-neural-networks-retail-case-study-example-part-8/	
Artificial Neural Networks – Retail Case Study Example In this case study, a classification problem to identify customers with a higher likelihood to purchase products from the campaign catalogues is discussed.	

Summary

- Fuzzy logic provides an alternative way to represent linguistic and subjective attributes of the real world in computing.
- Can be applied to control systems and other applications in order to improve the efficiency and simplicity of the design process.