Volume 5, Issue 12, December 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Multilayer Perceptron Algorithm (XOR) using Backpropagation

Nimisha Peddakam, Sreevidya Susarla, Annepally Shivakesh Reddy

CSE Department, CBIT, Telangana,
India

Abstract- A multilayer perceptron (MLP) is a feed forward artificial neural network model that maps sets of input data onto a set of appropriate outputs. A MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. Except for the input nodes, each node is a neuron (or processing element) with a nonlinear activation function.MLP utilizes a supervised learning technique called back propagation for training the network.MLP is a modification of the standard linear perceptron and can distinguish data that are not linearly separable. This project encompasses user friendly operations by using the tools from Matlab. The objective of this research is to implement both the single layer perceptron and multilayer perceptron algorithm for XOR problem.XOR(Exclusive or) is a logical operation that outputs true whenever both outputs differ(one is true, the other is false).Single layer perceptron takes less time but gives less accurate value. Multilayer perceptron overcomes this difficulty and results in a very accurate value but it takes time greater than the time taken by single layer perceptron. The philosophy of this project is to display the output pictorially by catering the hidden layers in the multilayer perceptron.

Keywords: Perceptron, Back propagation, sigmoid function, hidden layer

I. INTRODUCTION

The multilayer perceptron is a hierarchical structure of several perceptrons, andovercomes the disadvantage of single-layer networks. The multilayer perceptron is an artificial neural network that deals with nonlinear function mappings. The multilayer perceptron is capable of learning a rich variety of nonlinear decision surfaces.Nonlinear functions can be represented by multilayer perceptrons with units that use nonlinear transfer functions.

II. XOR PROBLEM AND MLP

The XOR problem is nothing but build a neural network that will produce the following truth table, called the 'exclusive or' or 'XOR' (either A or B but not both).

A	B	A XOR B
0	0	0
1	0	1
0	1	1
1	1	0

An MLP is a network of simple neurons called perceptrons. The basic concept of a single perceptron was introduced by Rosenblatt in 1958. The perceptron computes a single output from multiple real-valued inputs by forming a linear combination according to its input weights and then possibly putting theoutput through some nonlinear activation function. Mathematically this can be written as

$$y = \varphi(\sum_{i=1}^{n} w_i x_i + b) = \varphi(\mathbf{w}^T \mathbf{x} + b)$$

where **w** denotes the vector of weights, **x** is the vector of inputs, **b** is the bias and **Si** is the activation function. The original Rosenblatt's perceptron used a Heaviside step function as the activation function Nowadays, and especially in multilayer networks, the activation function is often chosen to be the logistic sigmoid f(x)= *1/(1*+e^{-x}). These functions are used because they are mathematically convenient and are close to linear near origin while saturating rather quickly when getting away from the origin. This allows MLP networks to model well both strongly and mildly nonlinear mappings.

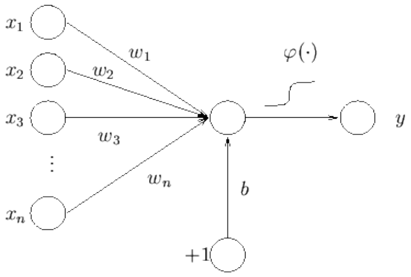


Fig 1 Signal-flow graph of the perceptron

A single perceptron is not very useful because of its limited mapping ability. No matter what activation function is used, the perceptron is only able to represent an oriented ridge-like function. The perceptrons can, however, be used as building blocks of a larger, much more structure. Atypical multilayer perceptron (MLP) network consists of a set of source nodes forming the input layer, one or more hidden layers of computation nodes, and an output layer of nodes. The input signal propagates through the network layer-by-layer. The computations performed by such a feed forward network with a single hidden layer with nonlinear activation functions and a linear output layer.

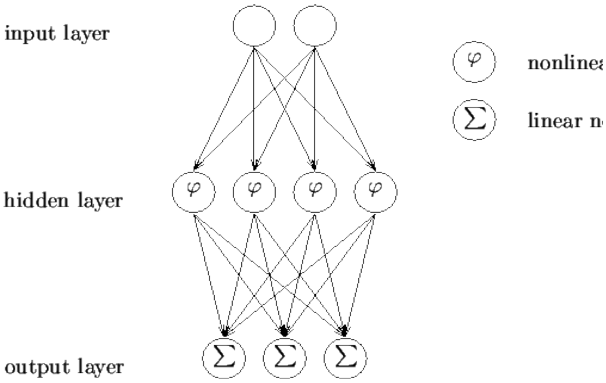


Fig 2 Signal-flow graph of an MLP

While single-layer networks composed of parallel perceptrons are rather limited in what kind of mappings they can represent, the power of an MLP network with only one hidden layer is surprisingly large and has continuous function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ to any given accuracy, provided that sufficiently many hidden units are available.MLP networks are typically used in supervised learning problems. This means that there is a training set of input-output pairs and the network must learn to model the dependency between them. The training here means adapting all the weights and biases (A, B, a and b in)to their optimal values for the given pairs (s(t),x(t)) . The criterion to be optimized is typically the squared reconstruction error.

$$\sum_t ||\mathbf{f}(\mathbf{s}(t)) - \mathbf{x}(t)||^2$$

The supervised learning problem of the MLP can be solved with the back-propagation algorithm. The algorithm consists of two steps. In the forward pass, the predicted outputs corresponding to the given inputs are evaluated. In the backward pass, partial derivatives of the cost function with respect to the different parameters are propagated back through the network. The chain rule of differentiation gives very similar computational rules for the backward pass as the ones in the forward pass. The network weights can then be adapted using any gradient-based optimization algorithm. The whole process is iterated until the weights have converged.

The MLP network can also be used for unsupervised learning by using the so called auto-associative structure. This is done by setting the same values for both the inputs and the outputs of the network. The extracted sources emerge from the values of the hidden neurons. This approach is computationally rather intensive. The MLP network has to have at least three hidden layers for any reasonable representation and training such a network is a time consuming process.

III. BACKPROPAGATION ALGORITHM

MLP utilizes a supervised learning technique called back propagation for training the network.MLP is a modification of the standard linear perceptron and can distinguish data that are not linearly separable.Back-propagation is one of the simplest and most general methods for training of multilayer neural networks. The power of back-propagation is that it enables us to compute an effective error for each hidden unit, and thus derive a learning rule for the input-to-hidden weights.Our goal now is to set the interconnection weights based on the training patterns and the desired outputs. Slow convergence speed- is disadvantage of error back-propagation algorithm.

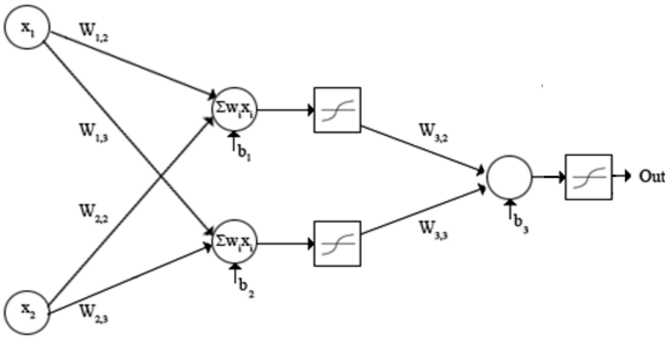


Fig 3 Neural network

Neural Networks are an information processing technique based on the way biological nervous systems, such as the brain, process information. They resemble the human brain in the following two ways:

- A neural network acquires knowledge through learning.
- A neural network's knowledge is stored within inter-neuron connection strengths known as synaptic weights.

Neural networks are being applied to an increasing large number of real world problems. Their primary advantage is that they can solve problems that are too complex for conventional technologies; problems that do not have an algorithmic solution or for which an algorithmic solution is too complex to be defined. In general, neural networks are well suited to problems that people are good at solving, but for which computers generally are not. These problems include pattern recognition and forecasting, which requires the recognition of trends in data.

Back propagation, an abbreviation for "backward propagation of errors", is a common method of training artificial neural networks used in conjunction with an optimization method such as gradient descent. The method calculates the gradient of a loss function with respect to all the weights in the network. The gradient is fed to the optimization method which in turn uses it to update the weights, in an attempt to minimize the loss function.Back propagation requires a known, desired output for each input value in order to calculate the loss function gradient. It is therefore usually considered to be a supervised learning method, although it is also used in some unsupervised networks such asautoencoders. It is a generalization of the delta rule to multi-layered feed forward, made possible by using the chain rule to iteratively compute gradients for each layer. Back propagation requires that the activation function used by the artificial neurons (or "nodes") be differentiable.

The true power and advantage of neural networks lies in their ability to represent both linear and non-linear relationships and in their ability to learn these relationships directly from the data being modeled. Traditional linear models are simply inadequate when it comes to modeling data that contains non-linear characteristics.The most common neural network model is the multilayer perceptron (MLP). This type of neural network is known as a supervised network because it requires a desired output in order to learn. The goal of this type of network is to create a model that correctly maps the input to the output using historical data so that the model can then be used to produce the output when the desired output is unknown.

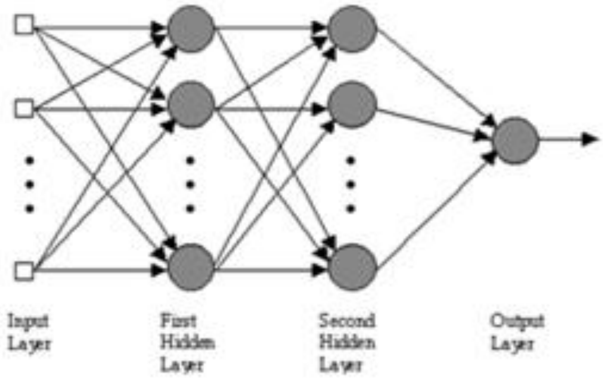


Fig 4 A graphical representation of an MLP

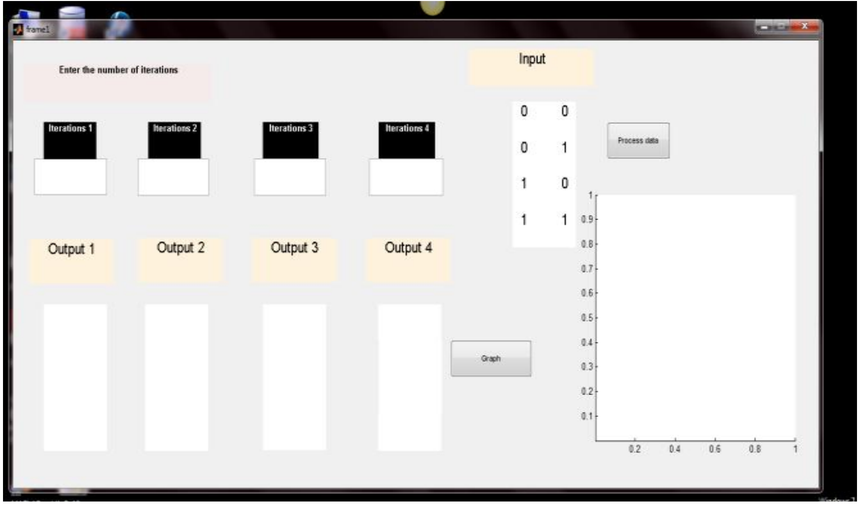
The MLP and many other neural networks learn using an algorithm called **back propagation**. With back propagation, the input data is repeatedly presented to the neural network. With each presentation the output of the neural network is compared to the desired output and an error is computed. This error is then fed back (back propagated) to the neural network and used to adjust the weights such that the error decreases with each iteration and the neural model gets closer and closer to producing the desired output. This process is known as "**training**".Neural networks have been successfully applied to broad spectrum of data-intensive applications, such as:

- **Process Modeling and Control** - Creating a neural network model for a physical plant then using that model to determine the best control settings for the plant.
- **Machine Diagnostics** - Detect when a machine has failed so that the system can automatically shut down the machine when this occurs.
- **Portfolio Management** - Allocate the assets in a portfolio in a way that maximizes return and minimizes risk.
- **Target Recognition** - Military application which uses video and/or infrared image data to determine if an enemy target is present.
- **Medical Diagnosis** - Assisting doctors with their diagnosis by analyzing the reported symptoms and/or image data such as MRIs or X-rays.
- **Credit Rating** - Automatically assigning a company's or individual credit rating based on their financial condition.
- **Targeted Marketing** - Finding the set of demographics which have the highest response rate for a particular marketing campaign.
- **Voice Recognition** - Transcribing spoken words into ASCII text.
- **Financial Forecasting** - Using the historical data of a security to predict the future movement of that security.
- **Quality Control** - Attaching a camera or sensor to the end of a production process to automatically inspect for defects.

- **Intelligent Searching** - An internet search engine that provides the most relevant content and banner ads based on the users' past behavior.
- **Fraud Detection** - Detect fraudulent credit card transactions and automatically decline the charge.

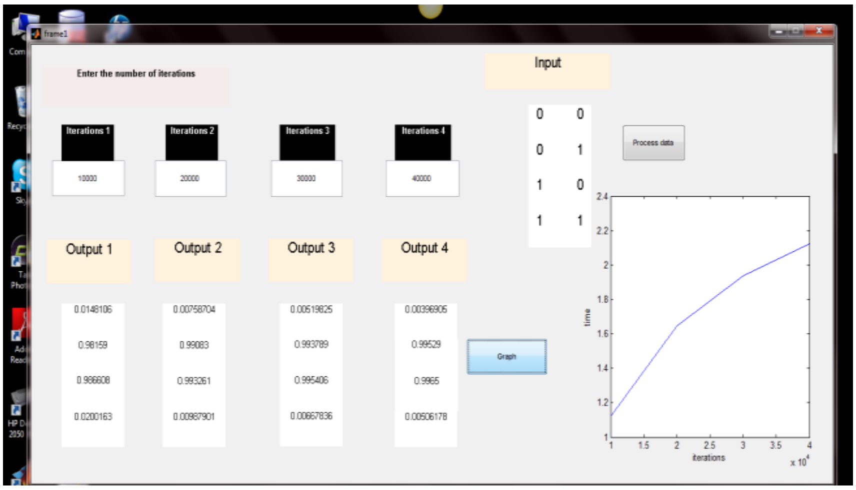
IV. RESULTS AND DISCUSSIONS

In multilayer perceptron, there are hidden layers between the input and the output. The weights are calculated by using the bias and then it generates the output. If the output that is produced in the hidden layers is greater than the input weight, then it is back propagated and the difference of the weights is then produced. Accurate weights are then produced, calculating the weights randomly.



Input	
0	0
0	1
1	0
1	1

Fig 5 Giving the input



Iterations 1	Iterations 2	Iterations 3	Iterations 4
10000	20000	30000	40000

Output 1	Output 2	Output 3	Output 4
0.0148106	0.00758704	0.00519825	0.00396905
0.98159	0.99083	0.993789	0.99529
0.986608	0.993261	0.995406	0.9965
0.0200163	0.00987901	0.00667836	0.00506178

Input	
0	0
0	1
1	0
1	1

Fig 6 Output after the training

REFERENCES

[1] The Neural networks and Deep learning website. [Online]. Available: http://neuralnetworksanddeeplearning.com/chap2.html

[2] The Wikipedia page on Backpropagation. [Online]. Available: https://en.wikipedia.org/wiki/Backpropagation

[3] The Mathworks website. [Online]. Available: http://in.mathworks.com/matlabcentral/answers/85249-sigmoid-function-value-for-parameter-a-and-b