**Back Propagation Algorithm**

**What is backpropagation?**

We can define the backpropagation algorithm as an algorithm that trains some given feed-forward Neural Network for a given input pattern where the classifications are known to us. At the point when every passage of the example set is exhibited to the network, the network looks at its yield reaction to the example input pattern. After that, the comparison done between output response and expected output with the error value is measured. Later, we adjust the connection weight based upon the error value measured.

Before we deep dive into backpropagation, we should be aware about who introduced this concept and when. It was first introduced in the 1960s and 30 years later it was popularized by David Rumelhart, Geoffrey Hinton, and Ronald Williams in the famous 1986 paper. In this paper, they spoke about the various neural networks. Today, backpropagation is doing good. Neural network training happens through backpropagation. By this approach, we fine-tune the weights of a neural net based on the error rate obtained in the previous run. The right manner of applying this technique reduces error rates and makes the model more reliable. Backpropagation is used to train the neural network of the chain rule method. In simple terms, after each feed-forward passes through a network, this algorithm does the backward pass to adjust the model’s parameters based on weights and biases. A typical supervised learning algorithm attempts to find a function that maps input data to the right output.  Backpropagation works with a multi-layered neural network and learns internal representations of input to output mappings.

**How does backpropagation work?**

Let us take a look at how backpropagation works. It has four layers: input layer, hidden layer, hidden layer II and final output layer.

So, the main three layers are:

1. Input layer
2. Hidden layer
3. Output layer

Each layer has its own way of working and its own way to take action such that we are able to get the desired results and correlate these scenarios to our conditions. Let us discuss other details needed to help summarizing this algorithm.

After choosing the weights of the network randomly, the backpropagation

algorithm is used to compute the necessary corrections. The algorithm can be

decomposed in the following four steps:

1. Feed-forward computation
2. Backpropagation to the output layer
3. Backpropagation to the hidden layer
4. Weight updates

The algorithm is stopped when the value of the error function has become

sufficiently small.

**Simple Functioning of the backpropagation approach:**

Input layer receives x

Input is modelled using weights w

Each hidden layer calculates the output and data is ready at the output layer

Difference between actual output and desired output is known as the error

Go back to the hidden layers and adjust the weights so that this error is reduced in future runs. This process is repeated till we get the desired output. The training phase is done with supervision.  Once the model is stable, it is used in production.

**Why do we need backpropagation?**

Backpropagation has many advantages, some of the important ones are listed below-

* Backpropagation is fast, simple and easy to implement
* There are no parameters to be tuned
* Prior knowledge about the network is not needed thus becoming a flexible method
* This approach works very well in most cases
* The model need not learn the features of the function

**Types of backpropagation**

There are two types of backpropagation networks.

* Static backpropagation
* Recurrent backpropagation

1. **Static backpropagation**

In this network, mapping of a static input generates static output. Static classification problems like optical character recognition will be a suitable domain for static backpropagation.

1. **Recurrent backpropagation**

Recurrent backpropagation is conducted until a certain threshold is met.  After the threshold, the error is calculated and propagated backward.

The difference between these two approaches is that static backpropagation is as fast as the mapping is static.

**Disadvantages of backpropagation**

* Input data holds the key to the overall performance
* Noisy data can lead to inaccurate results
* Matrix based approach is preferred over a mini-batch

In conclusion, Neural network is a collection of connected units with input and output mechanism, each of the connections has an associated weight. Backpropagation is the “backward propagation of errors” and is useful to train neural networks. It is fast, easy to implement and simple. Backpropagation is very beneficial for deep neural networks working over error prone projects like speech or image recognition.

**Recurrent networks**

The backpropagation algorithm can also be extended to the case of recurrent networks. To deal with this kind of systems we introduce a discrete time variable t. At time t all units in the network recompute their outputs, which are then transmitted at time t+1. Continuing in this step-by-step fashion, the system produces a sequence of output values when a constant or time varying input is fed into the network. A recurrent network behaves like a finite automaton.