

[90% Refund @Courses](#)[Trending Now](#)[Data Structures & Algorithms](#)[Foundational Courses](#)[Data Science](#)[Read](#)[Courses](#)[Jobs](#)

Backpropagation is an algorithm that backpropagates the errors from the output nodes to the input nodes. Therefore, it is simply referred to as the backward propagation of errors. It uses in the vast applications of neural networks in data mining like Character recognition, Signature verification, etc.

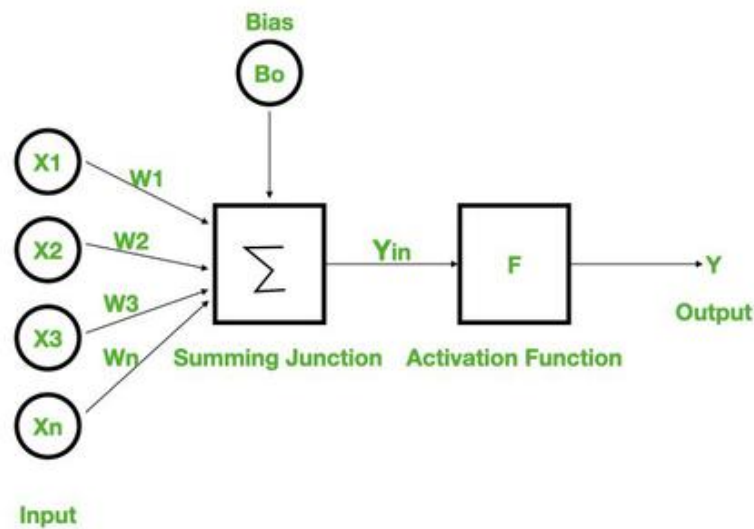
Neural Network:

Neural networks are an information processing paradigm inspired by the human nervous system. Just like in the human nervous system, we have biological neurons in the same way in neural networks we have artificial neurons, artificial neurons are mathematical functions derived from biological neurons. The human brain is estimated to have about 10 billion neurons, each connected to an average of 10,000 other neurons. Each neuron receives a signal through a synapse, which controls the effect of the signconcerning on the neuron.



We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !



Backpropagation:

Backpropagation is a widely used algorithm for training feedforward neural networks. It computes the gradient of the loss function with respect to the network weights. It is very efficient, rather than naively directly computing the gradient concerning each weight. This efficiency makes it possible to use gradient methods to train multi-layer networks and update weights to minimize loss; variants such as gradient descent or stochastic gradient descent are often used.

The backpropagation algorithm works by computing the gradient of the loss function with respect to each weight via the chain rule, computing the gradient layer by layer, and iterating backward from the last layer to avoid redundant computation of intermediate terms in the chain rule.



Features of Backpropagation:

1. it is the [gradient descent](#) method as used in the case of simple perceptron network with the differentiable unit.
2. it is different from other networks in respect to the process by which the weights are calculated during the learning period of the network.
3. training is done in the three stages :
 - the [feed-forward](#) of input training pattern
 - the calculation and backpropagation of the error
 - updation of the weight

Working of Backpropagation:

Neural networks use supervised learning to generate output vectors from input vectors that the network operates on. It Compares generated output to the desired output and generates an error report if the result does not match the generated output vector. Then it adjusts the weights according to the bug report to get your desired output.

Backpropagation Algorithm:

Step 1: Inputs X, arrive through the preconnected path.

Step 2: The input is modeled using true weights W. Weights are usually chosen randomly.

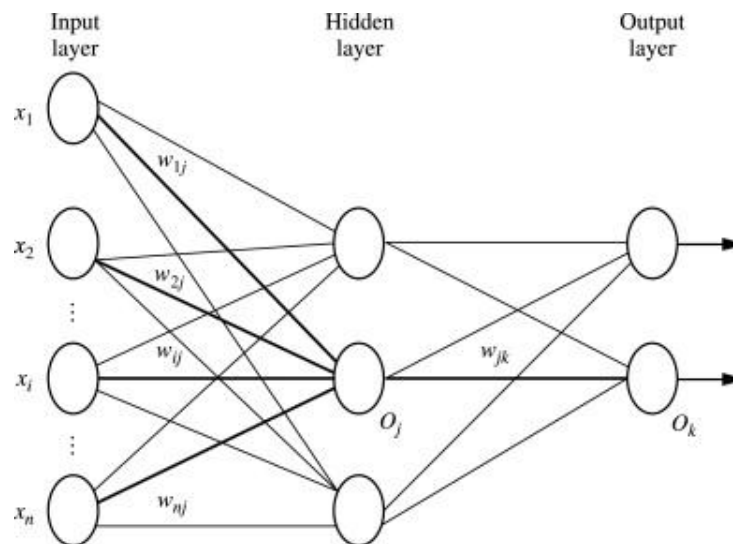
We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Step 4: Calculate the error in the outputs

Backpropagation Error= Actual Output – Desired Output

Step 5: From the output layer, go back to the hidden layer to adjust the weights to reduce the error.

Step 6: Repeat the process until the desired output is achieved.

**Parameters :**

- x = inputs training vector $x=(x_1, x_2, \dots, x_n)$.
- t = target vector $t=(t_1, t_2, \dots, t_n)$.
- δ_k = error at output unit.
- δ_j = error at hidden layer.
- α = learning rate.
- V_{0j} = bias of hidden unit j .

Training Algorithm :

Step 1: Initialize weight to small random values.

Step 2: While the stepsstopping condition is to be false do step 3 to 10.

Step 3: For each training pair do step 4 to 9 (Feed-Forward).

Step 4: Each input unit receives the signal unit and transmits the signal x_i

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Step 5 : Each hidden unit Z_j ($j=1$ to a) sums its weighted input signal to calculate its net input

$$z_{inj} = v_{0j} + \sum x_i v_{ij} \quad (i=1 \text{ to } n)$$

Applying activation function $z_j = f(z_{inj})$ and sends this signals to all units in the layer about i.e output units

For each output l =unit $y_k = (k=1 \text{ to } m)$ sums its weighted input signals.

$$y_{ink} = w_{0k} + \sum z_j w_{jk} \quad (j=1 \text{ to } a)$$

and applies its activation function to calculate the output signals.

$$y_k = f(y_{ink})$$

Backpropagation Error :

Step 6: Each output unit y_k ($k=1$ to n) receives a target pattern corresponding to an input pattern then error is calculated as:

$$\delta_k = (t_k - y_k) + y_{ink}$$

Step 7: Each hidden unit Z_j ($j=1$ to a) sums its input from all units in the layer above

$$\delta_{inj} = \sum \delta_j w_{jk}$$

The error information term is calculated as :

$$\delta_j = \delta_{inj} + z_{inj}$$

Updation of weight and bias :

Step 8: Each output unit y_k ($k=1$ to m) updates its bias and weight ($j=1$ to a). The weight correction term is given by :

$$\Delta w_{jk} = \alpha \delta_k z_j$$

and the bias correction term is given by $\Delta w_k = \alpha \delta_k$.

therefore $w_{jk(\text{new})} = w_{jk(\text{old})} + \Delta w_{jk}$

$$w_{0k(\text{new})} = w_{0k(\text{old})} + \Delta w_{0k}$$

for each hidden unit z_j ($j=1$ to a) update its bias and weights ($i=0$ to

n) the weight connection term

and the bias connection on term

$$\Delta v_{0j} = \alpha \delta_j$$

Therefore $v_{ij(\text{new})} = v_{ij(\text{old})} + \Delta v_{ij}$

$$v_{0j(\text{new})} = v_{0j(\text{old})} + \Delta v_{0j}$$

Step 9: Test the stopping condition. The stopping condition can be the minimization of error, number of epochs.

Need for Backpropagation:

Backpropagation is “backpropagation of errors” and is very useful for training neural networks. It’s fast, easy to implement, and simple. Backpropagation does not require any parameters to be set, except the number of inputs. Backpropagation is a flexible method because no prior knowledge of the network is required.

Types of Backpropagation

There are two types of backpropagation networks.

- **Static backpropagation:** Static backpropagation is a network designed to map static inputs for static outputs. These types of networks are capable of solving static classification problems such as OCR (Optical Character Recognition).
- **Recurrent backpropagation:** Recursive backpropagation is another network used for fixed-point learning. Activation in recurrent backpropagation is feed-forward until a fixed value is reached. Static backpropagation provides an instant mapping, while recurrent backpropagation does not provide an instant mapping.

Advantages:

- It is simple, fast, and easy to program.
- Only numbers of the input are tuned, not any other parameter.
- It is Flexible and efficient.

• No need for users to learn any special functions

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).