BACK PROPAGATION IN MACHINE LEARNING

Backpropagation is short for "backward propagation of errors." In the context of backpropagation, SGD**(Stochastic Gradient Descent)** involves updating the network's parameters iteratively based on the gradients computed during each batch of training data.

Instead of computing the gradients using the entire training dataset (which can be computationally expensive for large datasets), SGD computes the gradients using small random subsets of the data called mini-batches. Here's an overview of how backpropagation algorithm works:

1. **Forward Pass:** During the forward pass, input data is fed into the neural network, and the network's output is computed layer by layer. Each neuron computes a weighted sum of its inputs, applies an activation function to the result, and passes the output to the neurons in the next layer.

2. **Loss Computation:** After the forward pass, the network's output is compared to the true target values, and a loss function is computed to measure the discrepancy between the predicted output and the actual output.

3. **Backward Pass (Gradient Calculation):** In the backward pass, the gradients of the loss function with respect to the network's parameters (weights and biases) are computed using the chain rule of calculus. The gradients represent the rate of change of the loss function with respect to each parameter and provide information about how to adjust the parameters to decrease the loss.

4. **Parameter update:** Once the gradients have been computed, the network's parameters are updated in the opposite direction of the gradients in order to minimize the loss function. This update is typically performed using an optimization algorithm such as stochastic gradient descent (SGD), that we discussed earlier.

5. **Iterative Process:** Steps 1-4 are repeated iteratively for a fixed number of epochs or until convergence criteria are met. During each iteration, the network's parameters are adjusted based on the gradients computed in the backward pass, gradually reducing the loss and improving the model's performance.

The backpropagation algorithm is used in a Multilayer perceptron neural network to increase the accuracy of the output by reducing the error in predicted output and actual output.

According to this algorithm,

- Calculate the error after calculating the output from the Multilayer perceptron neural network.
- This error is the difference between the output generated by the neural network and the actual output. The calculated error is fed back to the network, from the output layer to the hidden layer.
- Now, the output becomes the input to the network.
- The model reduces error by adjusting the weights in the hidden layer.
- Calculate the predicted output with adjusted weight and check the error. The process is recursively used till there is minimum or no error.
- This algorithm helps in increasing the accuracy of the neural network.