Relationship between Backpropagation and Gradient Descent

Submitted by

Harsh

Registration Number – 201800469

Semester – 6th

Backpropagation

The backpropagation algorithm is used to effectively train a neural network through a method called chain rule. After each forward pass through a network, backpropagation performs a backward pass while adjusting the model's weight and biases.

Backpropagation repeatedly adjusts the weights of the connections in the network so as to minimize a measure of the difference between the actual output vector of the net and the desired output vector and the ability to create useful new features distinguishes back-propagation from earlier, simpler methods.

The level of adjustment of weights and biases is determined by the gradients of the cost function with respect to those parameters.

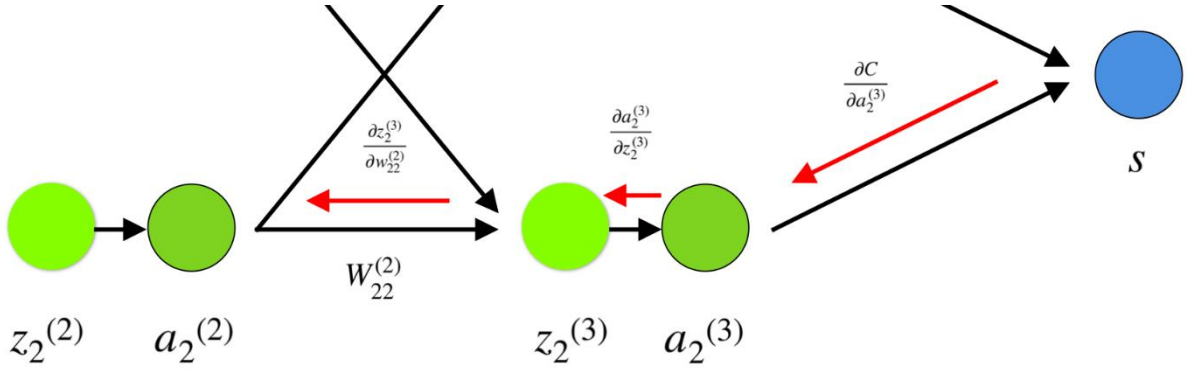


Figure 1: Visual representation of backpropagation in a neural network.

Gradient Descent

Gradient Descent is an optimizing algorithm used in Machine/ Deep Learning algorithms. The goal of Gradient Descent is to minimize the objective convex function f(x) using iteration.

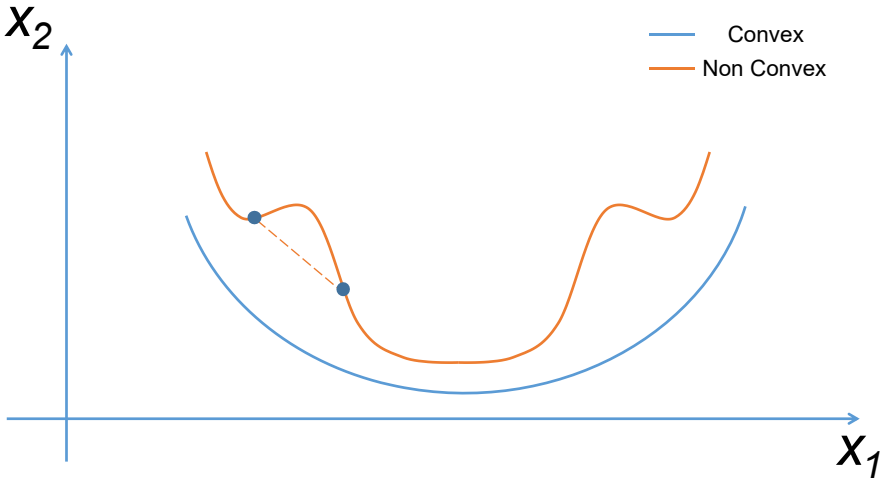


Figure 2: Convex function versus Non-Convex function.

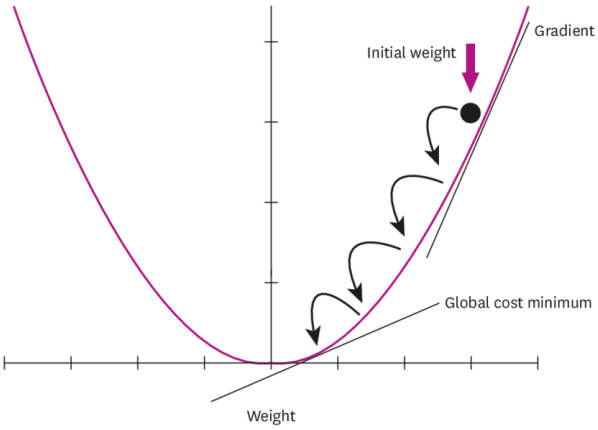


Figure 3: Intuition behind Gradient Descent.

A machine learning model always wants low error with maximum accuracy, which is achieved with Gradient descent.

In case of sparse data, we would experience sparse ON (1) features and more frequent OFF (0) features, now, most of the time gradient update will be NULL as derivative is zero in most cases and when it will be one, the steps would be too small to reach minima.

How does Gradient Descent and Backpropagation work together?

The derivative of a function gives the direction in which the function increases, and its negative, the direction in which the function decreases.

Training a model is just minimising the loss function, and to minimise we have to move in the negative direction of the derivative. Back-propagation is the process of calculating the derivatives and gradient descent is the process of descending through the gradient, i.e., adjusting the parameters of the model to go down through the loss function.

Back-propagation is called like this because to calculate the derivative we use the chain rule from the last layer (which is the one directly connected to the loss function, as it is the one that provides the prediction) to the first layer, which is the one that takes the input data. We are "moving from back to front".

In gradient descent one is trying to reach the minimum of the loss function with respect to the parameters using the derivatives calculated in the back-propagation. The easiest way would be to adjust the parameters by subtracting its corresponding derivative multiplied by a learning rate, which regulates how much you want to move in the gradient direction.

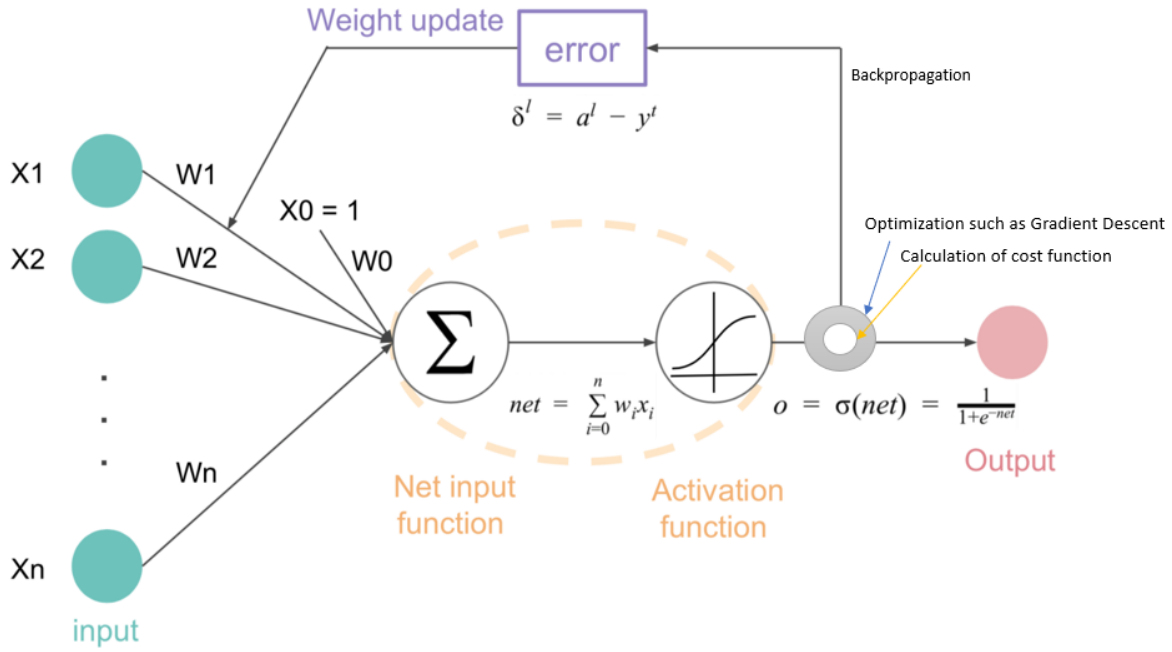
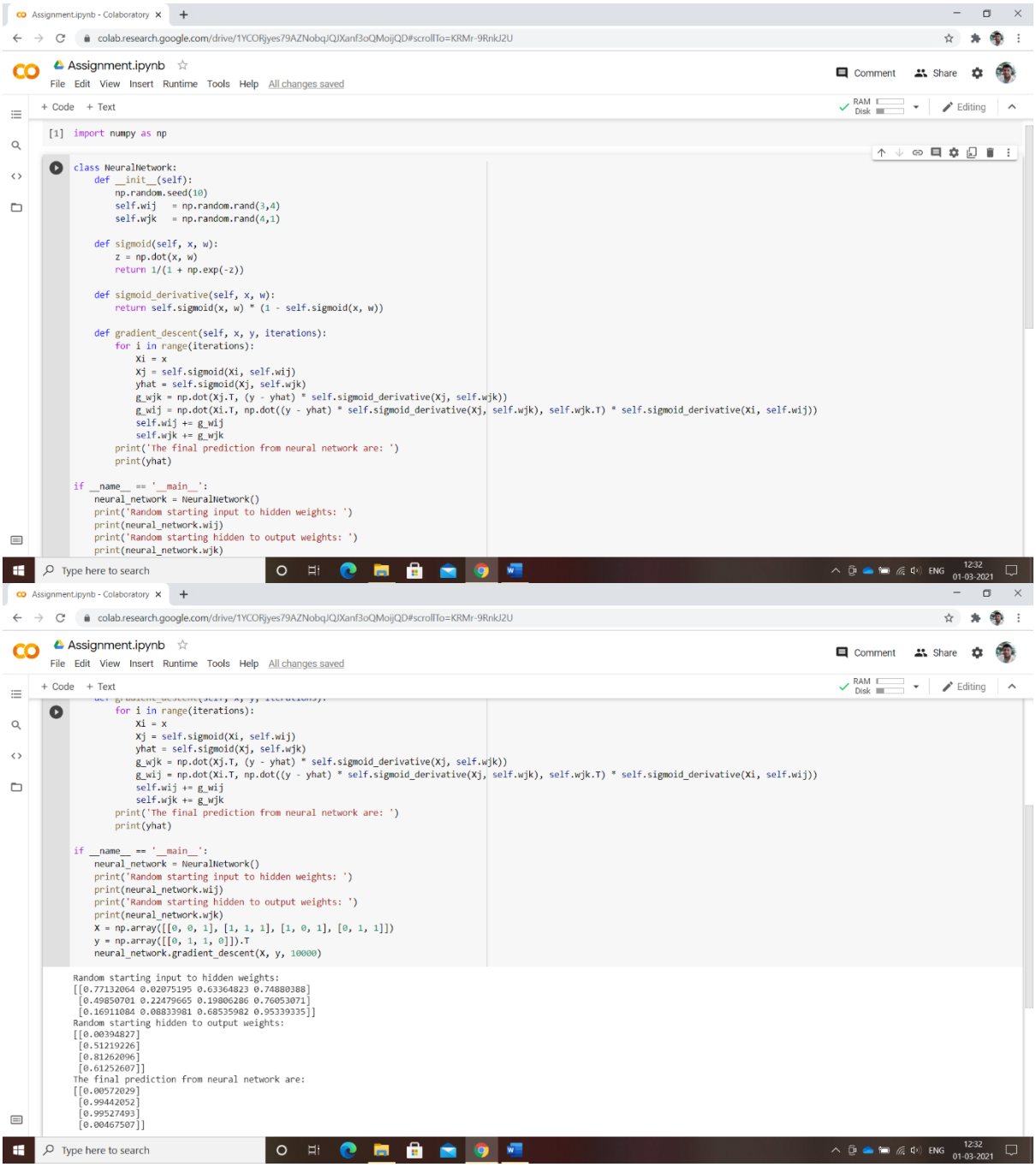


Figure 4: An understanding of the neural network and back-propagation.

An implementation of Gradient Descent and Backpropagation

- Used a small neural network with a single hidden layer only.
- Used the sigmoid activation function and assumed bias to be 0, for simplicity.
- Language used – Python 3
- Tool used – Google Colaboratory



Link to Project - https://github.com/writetoharsh/Relationship-between-Gradient-Descent-and-Backpropagation/blob/main/Assignment.ipynb

REFERENCES

1. Siemon Kostadinov, “Understanding Backpropagation Algorithm”, towardsdatascience.com, Aug.08, 2019 [Online]. Available: https://towardsdatascience.com/understanding-backpropagation-algorithm-7bb3aa2f95fd .
2. Daksh Trehan, “Gradient Descent Explained”, towardsdatascience.com, May.22, 2020 [Online]. Available: https://towardsdatascience.com/gradient-descent-explained-9b953fc0d2c .
3. Yitong Ren, “A Step-by-Step Implementation of Gradient Descent and Backpropagation”, towardsdatascience.com, May.30, 2019 [Online]. Available: https://towardsdatascience.com/a-step-by-step-implementation-of-gradient-descent-and-backpropagation-d58bda486110#:~:text=This%20is%20done%20using%20gradient,minima%20of%20the%20loss%20function. .