Gradient descent is an algorithm to find the minimum of a function. We would like to minimise the function to represent the minimum cost, which is the cost Function. Sometimes when the function cannot be differentiated easily, we use gradient descent to work out. Gradient descent is a method to tune the parameters of a function in order to reach the minimum error state; it will alter parameters until the cost function reaches minimum. Hence, gradient descent is very useful for machine learning and can be easily computed the minimum of cost function by training the machines.

For example, a traveller (neutral network) is lost in between mountains (cost function) the traveller is near by the mountain top and he is looking to go back to the lowest position to leave.

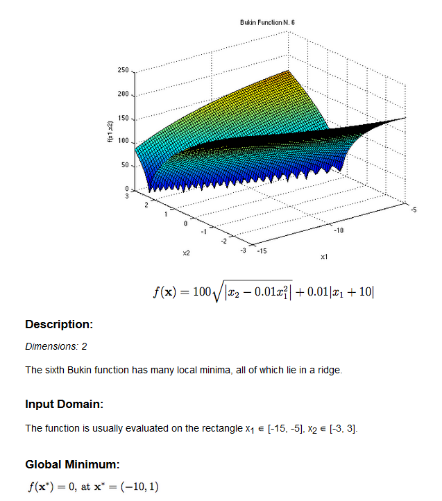
Measure the slope of the mountain (gradient) at the point and move one step (change in parameters) at a time going downhill.

Learning rate α controls the size of step during descent. If α is too small, the gradient descent will take too long to reach minima. If α is too big, the gradient descent may not converge.

Gradient descent is an update rule, it updates all parameters simultaneously. Which means the parameter is updated by subtracting the step value; apply update rule until the cost function reaches a stable minimum value.

However, the downfall is that there are local minimums and a global minimum, which the network may accidentally fall into a local minimum and gradient descent stops due to the error surfaces are not inherently convex.

Stochastic Gradient Descent takes a shuffled set of data, compute the derivative and make an update to the parameters for every point, the network would reach the minimum more efficiently.

Momentum adds a fraction of the past weight update to the current weight update leading the step size to change and alter its direction. This helps to prevent the network stuck in a local minima even the current gradient is zero. Hence movements along the error surface will be smoother and network can move more quickly throughout it.

All Vanilla Gradient Descents are running at 1000 iterations

Step size 0.00005



Step size 0.0001



Step size 0.0005



The most suitable step size is 0.001 with the smallest value of y suggesting an approx. global minimum

(-9.031, 0.816, 0.920).

Variant one – Stochastic Gradient Descent with step size 0.0001 and 10000 iterations



It illustrates a smaller value of y, hence it’s descending further more suggesting an approx. global minimum (-9.015, 0.813, 0.361).

Variant two – Momentum Gradient Descent with step size 0.0001 and 1000 iterations



Unfortunately, it shows a greater value of y than Stochastic Gradient Descent and Vanilla Gradient Descent for step size 0.0001 suggesting an approx. global minimum (-9.129, 0.833, 1.047)

In conclusion, Stochastic Gradient Descent with step size 0.0001 and 10000 iterations shows a point that is closest to the global minimum.