Gradient Descent

Thursday, April 7, 2022 7:44 AM

Gradient descent is one of the most popular algorithms to perform optimization and by far the most common way to optimize neural networks.

Gradient descent is a way to minimize an objective function $J(\theta)$ parameterized by a model's parameters $\theta \in Rd$ by updating the parameters in the opposite direction of the gradient of the objective function $\nabla\theta J(\theta)$ w.r.t. to the parameters. The learning rate η determines the size of the steps we take to reach a (local) minimum. In other words, we follow the direction of the slope of the surface created by the objective function downhill until we reach a valley.

Backpropagation Algorithm

epochs = 5

for i in range(epochs):

for j in range(X.shape[0]):

→ Select 1 row (random)

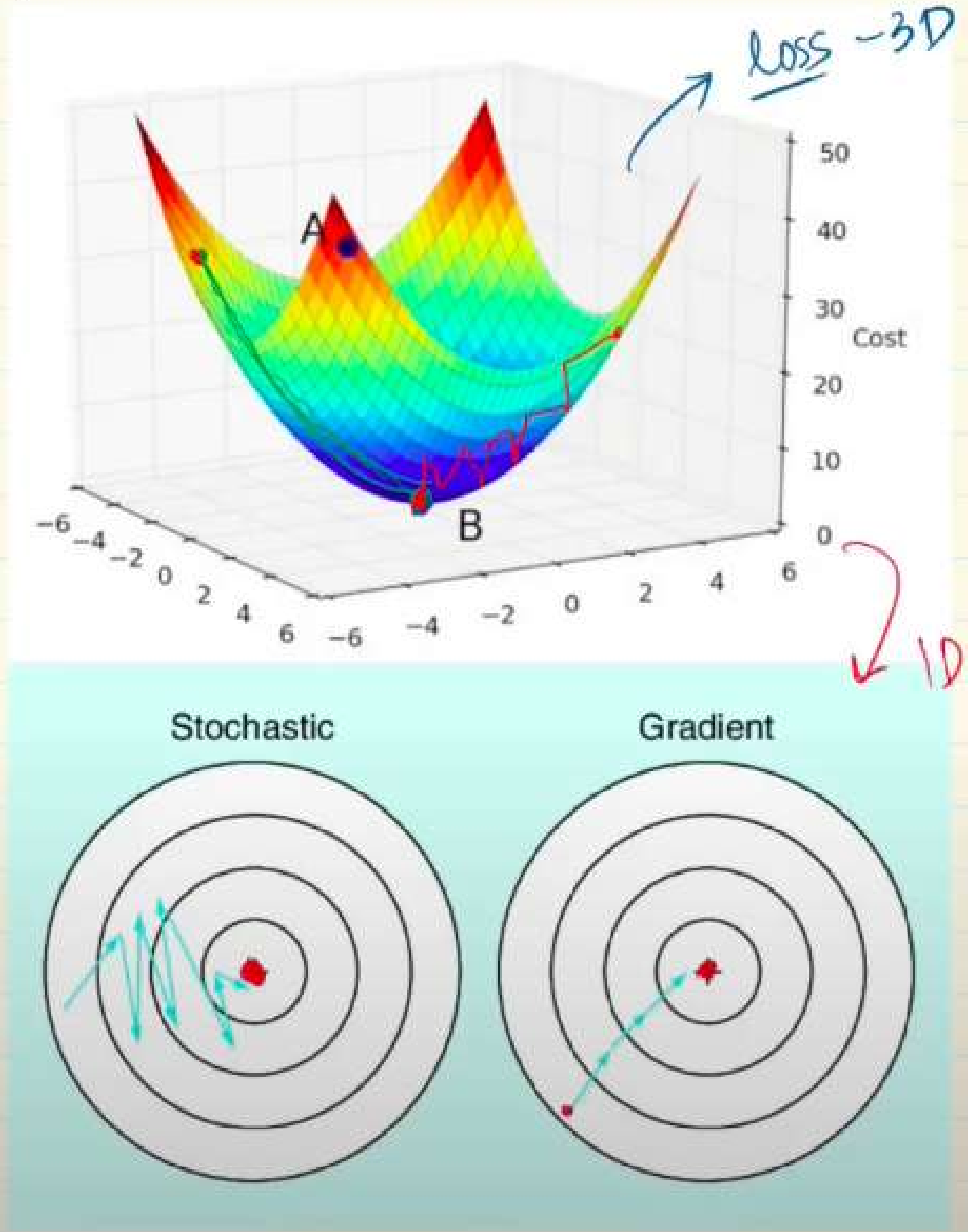
→ Predict (using Forward prop)

→ Calculate loss (using Loss function → mse)

→ Update weights and bias using GD

$$\boxed{W_n = W_0 - \eta \frac{\partial L}{\partial W}}$$

→ [Calculate avg loss for the epoch — L_{avg}]


loss -3D
A
B
Cost
50
40
30
20
10
0
-6 -4 -2 0 2 4 6
-6 -4 -2 0 2 4 6
1D
Stochastic
Gradient

Vanishing Gradient Problem

Thursday, April 7, 2022 7:45 AM

In machine learning, the **vanishing gradient problem** is encountered when training artificial neural networks with gradient-based learning methods and backpropagation. In such methods, during each iteration of training each of the neural network's weights receives an update proportional to the partial derivative of the error function with respect to the current weight. The problem is that in some cases, the gradient will be vanishingly small, effectively preventing the weight from changing its value. In the worst case, this may completely stop the neural network from further training. ~~As one example of the problem cause, traditional activation~~