Advanced Optimizers

Optimizers adjust model parameters iteratively during training to minimize a loss function, enabling neural networks to learn from data. Choosing an appropriate optimizer for a deep learning model is important as it can greatly impact its performance. Optimization algorithms have different strengths and weaknesses and are better suited for certain problems and architectures.

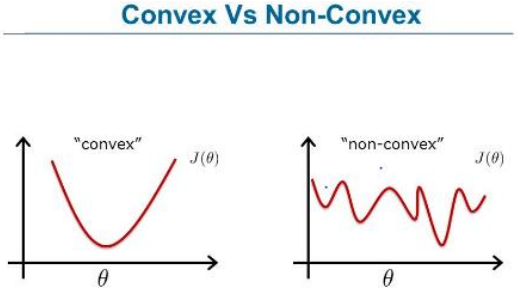
Some advanced optimizers used in neural networks:

1. SGD with Momentum
2. Nesterov Accelerated Gradient (NAG)
3. AdaGrad (Adaptive Gradient)
4. Gradient Descent with RMSprop(Root Mean Squared Propagation)
5. Adam (Adaptive Moment Estimation)

SGD with Momentum

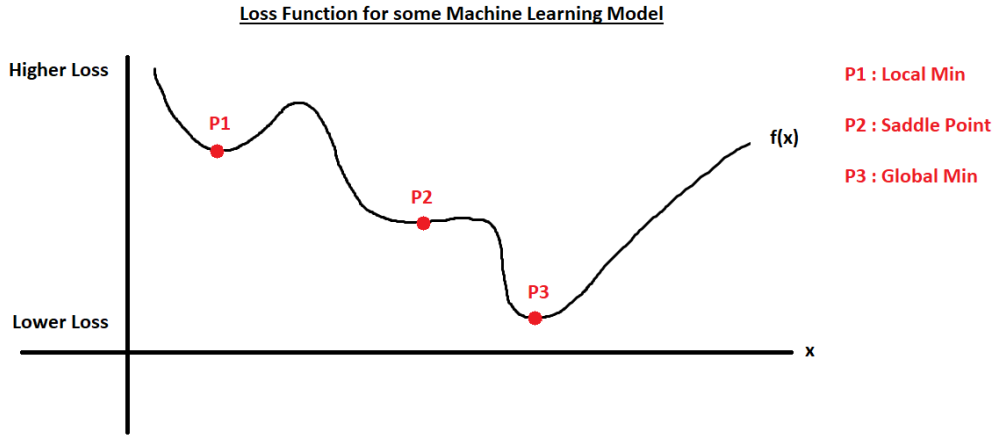
In machine learning, a cost function is a function that measures the error between the predicted and actual values. The goal is to minimize this error to improve the performance of the model.

A cost function is said to be convex if it is shaped like a bowl, with a single minimum point. On the other hand, a non-convex cost function has multiple local minimum points, and the global minimum may not be easily identifiable.



In the SGD we have some issues in which the SGD does not work perfectly because in deep learning we got a non-convex cost function graph and if use the simple SGD then it leads to low performance. There are 3 main reasons why it does not work:

- local minima and not able to reach global minima
- Saddle Point will be the stop for reaching global minima
- High curvature



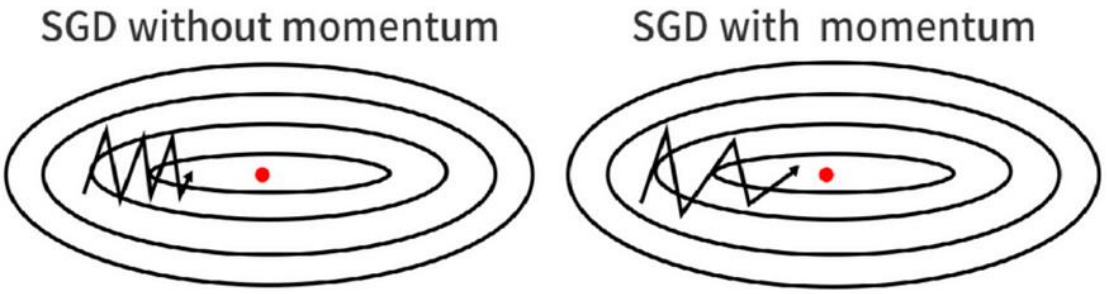
SGD with Momentum:

SGD Momentum is one of the optimizers which is used to improve the performance of the neural network. The term velocity v is used to denote the change in the velocity of the gradient to get to the global minima. The change in the weights is denoted by the formula:

$$W_{t+1} = W_t - V_t$$

$$here, V_t = \beta * V_{t-1} + \eta \Delta W_t$$

The past velocity for calculating Vt we have to calculate Vt-1 and for calculating Vt-1 we have to calculate Vt-2 and likewise. So we are using the history of velocity to calculate the momentum and this is the part that provides acceleration to the formula.



Here we have to consider two cases:

1. β=0 then, as per the formula weight updating is going to just work as a Stochastic gradient descent. Here we called β a decaying factor because it is defining the speed of past velocity.
2. β=1 then, there will be no decay. It involves the dynamic equilibrium which is not desired so we generally use the value of β like 0.9,0.99or 0.5 only.

Advantages of SGD with Momentum :

1. Momentum is faster than stochastic gradient descent the training will be faster than SGD.
2. Local minima can be an escape and reach global minima due to the momentum involved.

Disadvantages of SGD with Momentum :

But there is a catch, the momentum itself can be a problem sometimes because of the high momentum after reaching global minima it is still fluctuating and take some time to get stable at global minima. And that kind of behavior leads to time consumption which makes SGD with Momentum slower than other optimization out there but still faster than SGD.