Optimizers

Neural Network is Optimization Problem jaha loss minimize krna chahte hai

Loss -> Actual – Predicted

Role of Optimizer

Neural Network mai jo Weights & Biases ke value hai usiko sbhse optimum banate hai

At start Shuru mai Random value lete hai Weights & Biases ka

Random value se slowly sahi value tak pochneka try karte hai

Random Value se start krke sbhse optimum value jaha loss sbhse kamm hai udr pochna chahte hai & vaha ka Weights uthayenge fix krenge

Gradient Descent mai Simple Update rule use krte hai

Weight New = Weight Old – Learning Rate of Derivative of Loss / Derivate of weight Old.

Yeahi hum Loop mai Multiple Epochs baar chalate hai.

Gradient Descent mai 3 Types hote hai

Batch GD

Stochastic GD

Mini Batch GD

Batch GD

ismai one go mai saare data pe Ghumte hai

E.g:

500 rows ka data hai , 500 rows pe Prediction nikalte hai , Loss calculate krte hai then Weight Update krte hai.

SGD

SGD mai har data point ko dekhnekebaad update krte hai

E.g:

Mtlb 500 rows hai tou 500 x10 mtlb 5thousand baar Weights Update karenge.

Mini Batch

Yaha batch size decide krte hai let take batch_size = 100

Then 1 batch kebaad hie weight update hoga mtlb

If epoch 10 hai then 100 ka batch & 500 rows

*Mtlb 10*5 = 50 times*

Challenges of Gradient Descent

1. ***Deciding Right Value of Learning Rate***
   Agar LR is very Small then Convergence will be very small & if we have taken Large Value then Minima se hie Far away chale jaaye.

Solution of this problem was (Learning Rate Scheduling)

Threshold jo decide hota hai that we have to decide training se pehle.

2. ***Multiple Weights ka Value Nikalna hai***
   E.g 9 Features hai then 9D mai move karke Minima pr Reach karna hai.
   Learning Rate same Rate se speed pe chalta hai
   E.g 9 weights hai tou sbkeliye Same Learning Rate se neeche chalega.

3. *Problem of Local Minima*

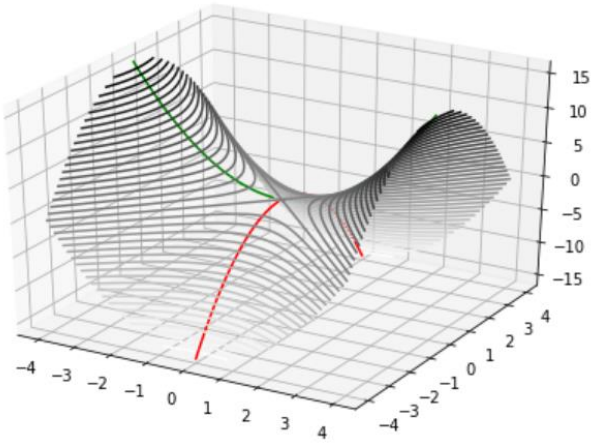
Multiple Minima Se nikalneka issue

SGD mai Potential hai firbhi ki Local Minima se Nikaljaye but baki mai nahi hai

4.) Saddle Point

David Li-Bland's Blog - Saddle Points and Stochastic Gradient Descent (davidlibland.github.io)

We now introduce the first villain in our saga, saddle points, which are known to cause problems for gradient descent. A saddle point is a critical point[1] of a function which is neither a local minima or maxima.



Gradient descent proceeds extremely slowly near a saddle point.

a saddle point is a critical point of a function where the gradient is zero, but it is not a local minimum or maximum. Saddle points can pose challenges for optimization algorithms like gradient descent because the gradient is zero at these points, and traditional gradient descent may struggle to escape from them.

Gradient descent is an iterative optimization algorithm used to minimize a function by adjusting its parameters. At each iteration, the algorithm moves in the direction opposite to the gradient of the function with respect to the parameters. This helps it converge towards a local minimum.

However, when the algorithm encounters a saddle point, the gradient is zero, and the descent direction becomes ambiguous. In some dimensions, the function may be increasing, while in others, it may be decreasing. This leads to slow convergence or even getting stuck at the saddle point.

Yeahisbh Reasons ke Vajese Hum aur Optimizers padhte hai.

Different Types of Optimizers

1.) Gradient Descent
2.) SGD (Stochastic Gradient Descent)
3.) Mini Batch Gradient Descent
4.) SGD with Momentum
5.) Adagrad (Adaptive Gradient)
6.) NAG
7.) RMS
8.) Adam optimizer (Currently the Best Optimizer in the Market)

Batch, Epochs, Iterations

1.) Gradient Descent

Weight New = Weight Old – Learning Rate of Derivative of Loss / Derivate of weight Old.

What is Epoch?

Whenever I do 1 Forward propagation & 1 Backward Propagation

→ **Forward**

← **Backward**

it is consider as 1 Epoch