Optimizers :

- Optimizers are algorithms or methods used to adjust the weights and biases of neural networks in order to minimize the loss function.
- The goal of optimization in machine learning is to find the best set of parameters (weights) that result in the best performance (lowest error or loss).

01. Momentum SGD
02. AdaGrad
03. RMSprop
04. Adam
05. Gradient Descent
06. Stochastic Gradient Descent
07. Mini batch Gradient Descent

-Why use Of Optimizers

1. Parameter Adjustment

- Description: Optimizers adjust the weights and biases of the neural network to minimize the loss function, ensuring the model learns effectively from the data.

2. Convergence Speed

- Description: Optimizers affect how quickly the training process converges to the minimum loss. Advanced optimizers like Adam and RMSprop can significantly speed up convergence compared to basic methods like standard SGD.

3. Avoiding Local Minima

- Description: Some optimizers help in escaping local minima and saddle points in the loss landscape. Techniques like momentum and adaptive learning rates are particularly useful in this regard.

Gradient Descent:

- Gradient descent is an optimization algorithm based on a convex function and tweaks its parameters iteratively to minimize a given function to its local minimum.
- Gradient Descent iteratively reduces a loss function by moving in the direction opposite to that of steepest ascent.

Learning Rate :

How big/small the steps are gradient descent takes into the direction of the local minimum are determined by the learning rate, which figures out how fast or slow we will move towards the optimal weights.

Stochastic Gradient Descent :

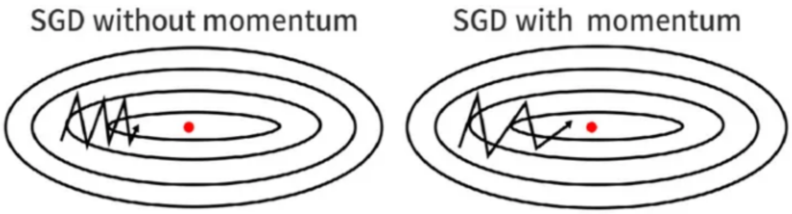
It is a variant of Gradient Descent. It update the model parameters one by one. If the model has 10K dataset SGD will update the model parameters 10k times.

Mini-Batch Gradient Descent :

It is a combination of the concepts of SGD and batch gradient descent. It simply splits the training dataset into small batches and performs an update for each of those batches.

SGD with Momentum :

SGD with Momentum is a stochastic optimization method that adds a momentum term to regular stochastic gradient descent.



AdaGrad(Adaptive Gradient Descent)

In all the algorithms that we discussed previously the learning rate remains constant. The intuition behind AdaGrad is can we use different Learning Rates for each and every neuron for each and every hidden layer based on different iterations.

Data Sparsity refers to the condition where a large percentage of data within a dataset is missing or is set to zero.

RMS-Prop (Root Mean Square Propagation)

RMS-Prop is a special version of Adagrad in which the learning rate is an exponential average of the gradients instead of the cumulative sum of squared gradients.

Adam(Adaptive Moment Estimation)

Adam optimizer is one of the most popular and famous gradient descent optimization algorithms. It is a method that computes adaptive learning rates for each parameter. It stores both the decaying average of the past gradients , similar to momentum and also the decaying average of the past squared gradients , similar to RMS-Prop and Adadelta.

How to choose optimizers?

- If the data is sparse, use the self-applicable methods, namely Adagrad, Adadelta, RMSprop, Adam.
- RMSprop, Adadelta, Adam have similar effects in many cases.
- Adam just added bias-correction and momentum on the basis of RMSprop,
- As the gradient becomes sparse, Adam will perform better than RMSprop.