Optimizers

What are Optimizers ?

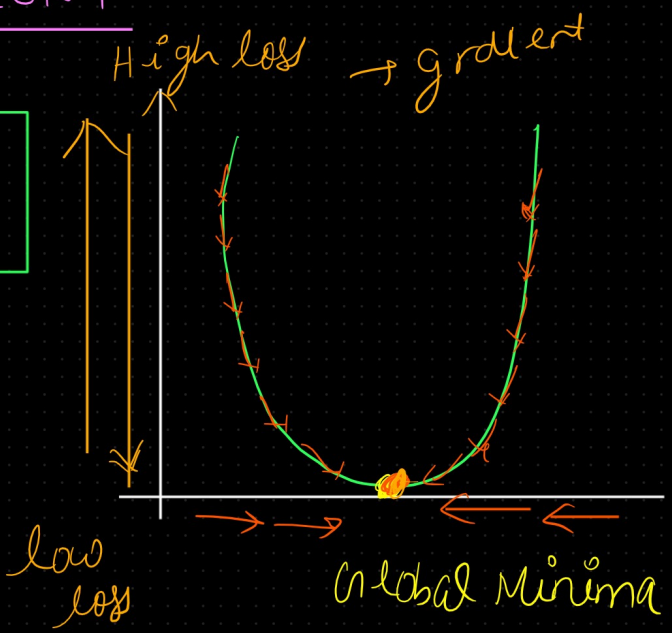
Optimizers are algorithms used in training deep learning models to minimize the loss function and improve the model's performance.

Some common optimizer functions :-

1. Gradient Descent (GD)
2. Stochastic Gradient Descent (SGD)
3. Min - Batch Gradient Descent
4. Momentum
5. Adagrad
6. RMSprop
7. Adam (Adaptive Moment Estimation)

(1) GRADIENT DESCENT

$$W_{new} = W_{old} - \eta \frac{\partial L}{\partial W_{old}}$$



GD → GD is a first-order iterative optimization algorithm for finding the minimum of a function.

Intution

It works by iteratively moving in the direction of the negative gradient of the loss function with respect to the model parameters.

(weight, bias)

↓

gradient descent

gradient

↓

Input → vector

↓

gradient — dir^, mag.

dir^ → ↑↑

{1000000}

⇓

gradient

FP →

Epoch

← BP

1 Epoch → pass

Weight update

Minimum loss function

Machine → High Conf.

1000 000 → gradient

FP →

1 Epoch

← BP

1 Epoch pass

① RAM ↑↑
GPU,

smooth loss fun^n
& convex ↑↑

Disadvantage

↓ Huge RAM

Resource Extensive

② Stochastic Gradient Descent (SGD)

entire dataset

→ GD → 1000000 → gradient

↑ ⟷ epoch

SGD → 1000000 → 1 → gradient

Iteration

⟷ 2 — gradient

3 — gradient

1000 000

1 Epoch

1 record ⟷ $\hat{y}$ 1st

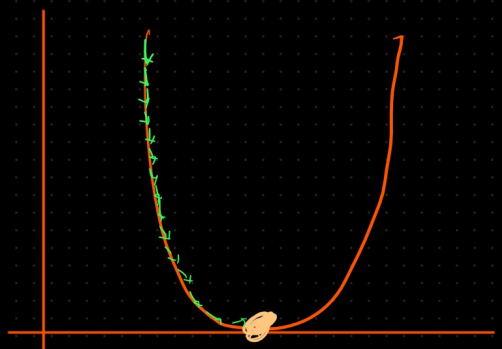
weight update

2 record ⟷ $\hat{y}$ 2nd

⋮

⟷ million + train

SGD is an extension of GD where instead of computing the gradient of the entire dataset, it computes the gradient of the loss for each training example and updates the parameters accordingly.



Advantage

1) RAM ↓ ✗

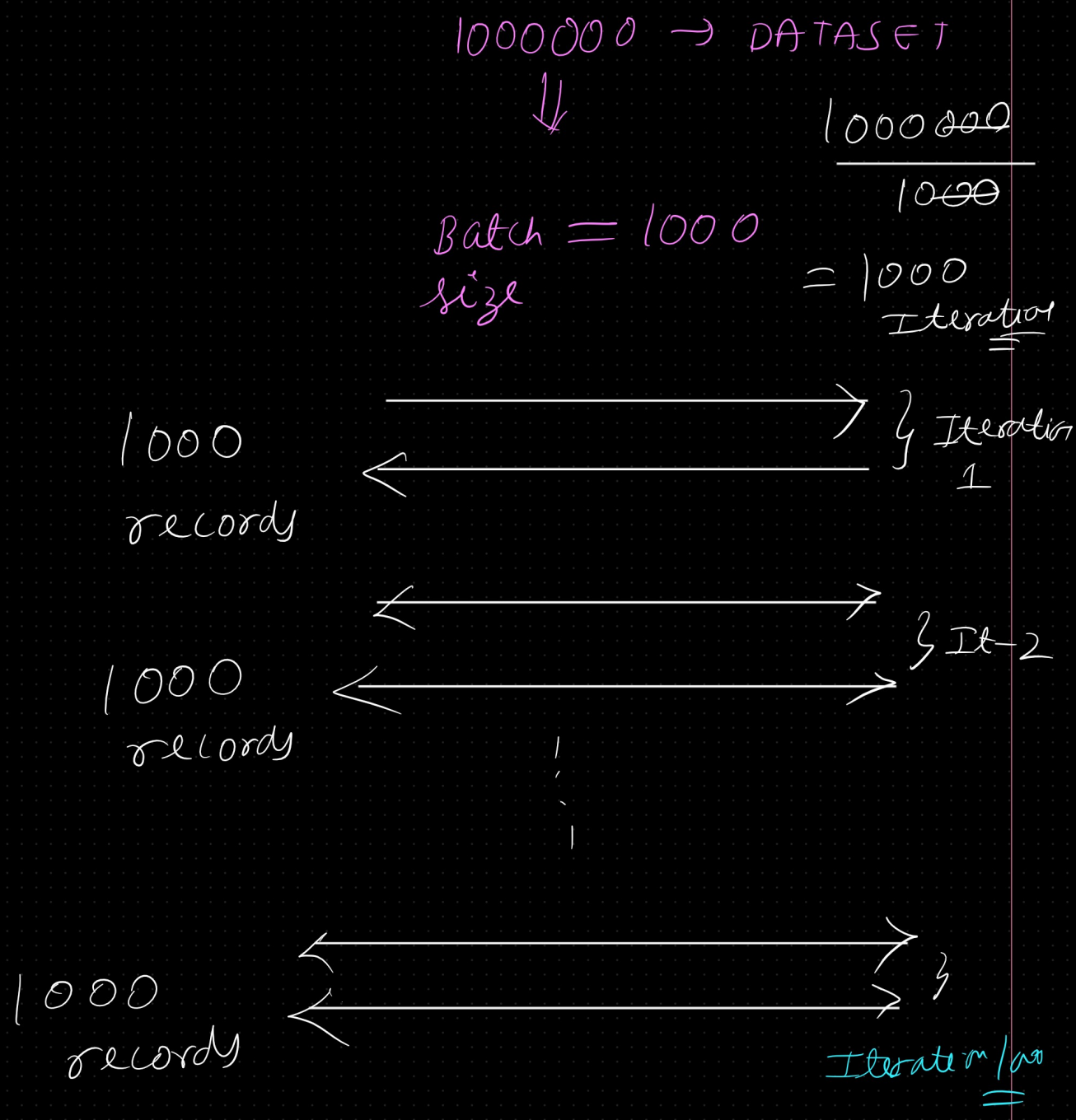
2) More computationally Efficient than GD

Disadvantage

1) convergence will be very slow

2) time complexity ↑↑

3) Mini Batch SGD



Mini-batch GD is a compromise between GD and SGD. It divides the dataset into small batches and computes the gradient of the loss for each batch.

Advantage

① Faster convergence than SGD

Disadvantage

1. Resource Intensive
2. Require tuning the Batch size which affect the convergence and memory usage

(4) Momentum → Smooth

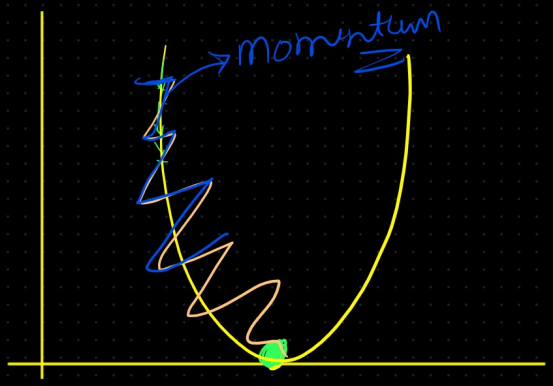
time series data

Weight updation

$$w_{new} = w_{old} - \eta \frac{\partial L}{\partial w_{old}}$$

$$w_t = w_{t-1} - \eta \frac{\partial L}{\partial w_{t-1}}$$

Momentum is a technique that accelerates SGD by adding a fraction of the update vector of the past time step to the current update vector.



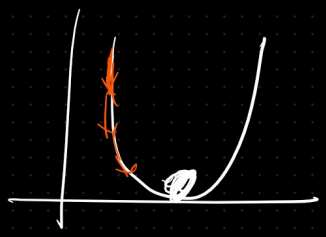
It helps to smooth out variations in the gradient descent path, especially in the presence of noisy gradients or high curvature.

(5) Adagrad

α - learning Rate → Adaptive

Adagrad adapts the learning rate of each parameter based on the historical gradients for that parameter.

Intuition



$\frac{\partial l}{\partial w}$ = gradient ↑ = α → scale down

gradient ↓↓ = α - scale up

Disadvantage

Accumulates the squared gradients in the denominator, leading to diminishing learning rates over time.

(6) RMSProp

RMSprop is an adaptive learning rate optimization algorithm that divides the learning rate by an exponentially decaying average of squared gradients.

$$\eta' = \frac{\eta}{\text{Exponential}}$$

$$\eta' = \frac{\eta}{\sqrt{\sigma_{t+1} + \epsilon}}$$

It scales the learning rate differently for each parameter based on the magnitudes of recent gradients.

(7) Adam (Adaptive Moment Estimation)

Best

Adam combines the ideas of momentum and RMSprop. It computes adaptive learning rates for each parameter and keeps track of both the first and second moments of the gradients.

It is less sensitive to the choice of hyperparameters and can handle sparse gradients more effectively.

Advantage

1) Fast Convergence
2) Good Generalisation

Disadvantages

Introduces additional hyperparameters (momentum coefficients and decay rates) that need to be tuned.

May exhibit erratic behavior on certain types of problems or architectures if hyperparameters are not properly adjusted.