Day-3.

* Ophimizers.
  1. Gradient Decent
  2. SAD (stochastic Gradient D).
  3. Mini Batch SAD
  4. SGD with momentum
  5. Adagrad.
  6. RMSPROP
  7. Adam ophimizers.

* Batch
* Epochs.
* Iterations.

} ANN.

1. Gradient Decent ← ophimizers.

* Disadvantage of Gradient Decent

1. Resource ~~Ex~~tensive (Huge Ram)

* Stochastic Gradient Decent

1 record. ⟶ ⟵ } → Iteration 1.

2 record. ⟶ ⟵ } → Iteration 2

* Disadvantage of Stochastic Gradient Decent.

① Convergence will be very slow.

* Mini Batch (SGD).

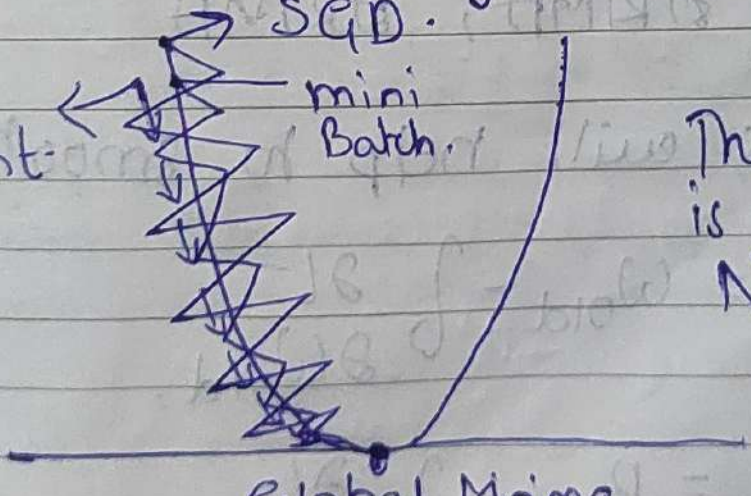
Epoch 1 ⟶ } Iteration 1.
10000 ⟵

Epoch 2 ⟶ } Iteration 2.
⟵

⋮

} → Iteration. 3.

→ Resources Intensive.
→ Convergences will be better.
→ Time Complexity will improve.



* In Mini Bated the Zigzag moement will be le.
* In SGD the Zigzag will be high.

There is a Noise

How do we remove the noise ??

→ We use a concept which is called as momentum.

④ SGD with momentum

→ What is this momentum will do is that it will smoothen this journey

{ Exponential Weight Average }.

⇓

Time Series.

⇓

ARIMA, ARMA.

This will help to smoothen

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

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

$\eta \frac{\partial L}{\partial \omega_{old}}$

ω = weights.
t = Time.

$$\omega_t = \omega_{t-1} - \eta \frac{\partial L}{\partial \omega_{t-1}}$$

* Exponential weights Average.

Dataset:

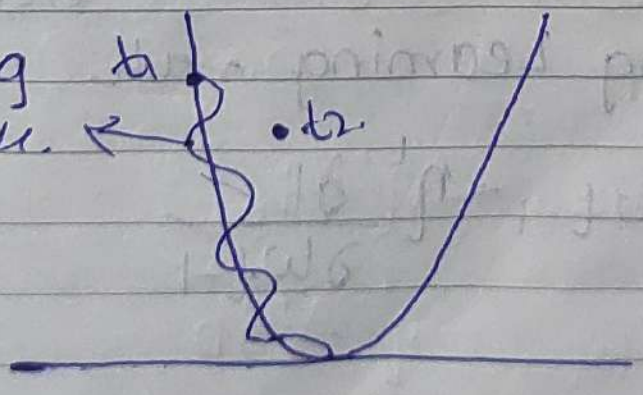
t_1	t_2	t_3	t_4	.....	t_n
a_1	a_2	a_3	a_4	.....	a_n

V = Value. β = Beta. if $\beta = 0.95$

$V_{t_1} = a_1$

$V_{t_2} = \beta * V_{t_1} + (1-\beta) * a_2$

$= (0.95) * V_{t_1} + (0.05) * a_2$

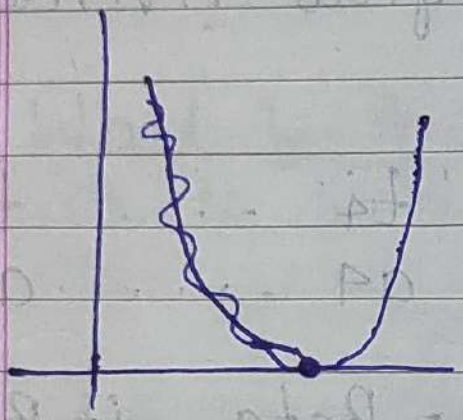


Exponential weight Avg.

$$w_t = w_{t-1} - \eta\, V_{dw}$$

$$V_{dw_t} = \beta \times V_{dw_{t-1}} + (1-\beta) * \frac{\partial L}{\partial w_{t-1}}$$

* The Problem we are solving.



1. Reducing the Noise.
2. Mini Batch.
3. Quick Convergence.

* Adagrad → Adaptive Gradient Decent.

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

η = fixed ⇓ Adaptive

* Changing Learning rate.

$$w_t = w_{t-1} - \eta' \frac{\partial L}{\partial w_{t-1}}$$

ϵ = Epsilon.

$$\eta' = \frac{\eta}{\sqrt{\alpha_t + \epsilon}}$$

ϵ → Small number.

denominator never become 0

* Our main aim is to keep decreasing as we ~~reach~~ reach Global minima.

t = Current time-stamp

$$\alpha_t = \sum_{i=1}^{t} \left(\frac{\partial L}{\partial w_t}\right)^{\uparrow\uparrow\uparrow}$$

t_1	t_2	t_3
$\eta = 0.01$	0.05	0.02

* As we reach near global minima this value will keep on decreasing that why we are bringing Adaphiveness in the learning rate.

* That why the learning rate never be fixed. it will be decreasing as we move towards global minima.

α_t = will be a Huge Number.

alpha does not reaches a Huge value.
to prevent this.

↓

* Adadelta & RMSProp.

$$\eta' = \frac{\eta}{\sqrt{sdw + \epsilon}}$$

$$sdw = \beta \cdot sdw_{t-1} + (1-\beta)\left(\frac{\partial L}{\partial w_{t-1}}\right)^2$$

(EWA)

$$\beta = 0.95$$

$$sdw_L = (0.95)\, sdw_{t-1} + (0.05)\left(\frac{\partial L}{\partial w_{t-1}}\right)^2$$

* This will always increases with the smalles number because of Beta value.

* Adam

In Adam Optimizer we combine momentum along with the RMSprop it becomes Adam Opt

$$wt = w_{t-1} - \eta' Vdw$$

$$bt = w_{bt-1} - \eta' Vdb$$

$$V dw_t = \beta \times V dw_{t-1} + (1-\beta) \frac{\partial L}{\partial w_{t-1}}$$

$$V db_t = \beta \times V db_{t-1} + (1-\beta) \frac{\partial L}{\partial b_{t-1}}$$

* It is solving the problem of :

① Smoothing.

② learning rate becomes Adaptive.