Tutorial-3

Optimizers and ANN

Day-3 — Deep Learning.

Agenda

① Optimizers — (i) Gradient Descent

(ii) SGD (Stochastic Gradient Descent)

(iii) Mini Batch SGD

(iv) SGD with momentum

(v) Adagrad (Adaptive Gradient Descent)

(vi) RMSPROP

(vii) Adam Optimizer

What is Batch, Epochs, Iterations

↓

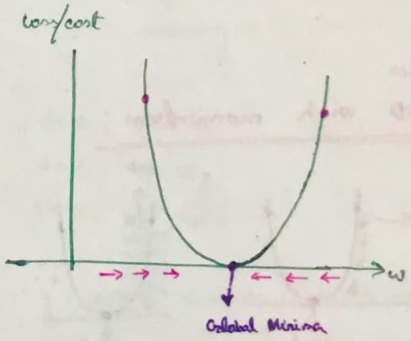
ANN

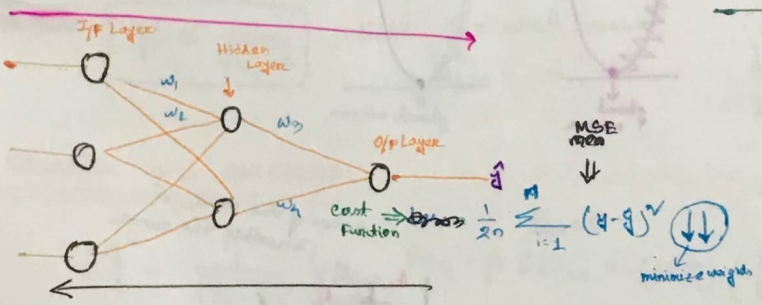
① Gradient Descent:

Weight updation formula-

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

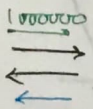
(η: learning rate)





optimizers

Epoch



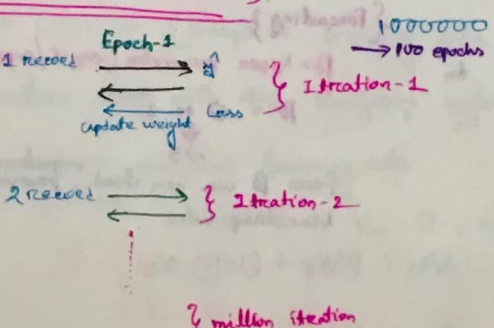
it means that {1000000} 1 F/W and B/W propogation.

Combination of forwards and backwards propogation.

Disadvantage

① Resource Extence (Huge RAM)

(ii) SGD (Stochastic Gradient Descent):



① Ram ↓↓

Disadvantages:

* Convergence will be very very slow
② Time complexcity will be very high

Mini Batch SGD:

1000000 , batch_size = 1000

∴ iteration = $\frac{\text{record}}{\text{batch size}}$ = $\frac{1000000}{1000}$ = 1000

Epoch-1 1000 ⟹ } iteration-1

Epoch-2 1000 ⟹ } iteration-2

} iteration-1000

1st iteration's = 1000 record

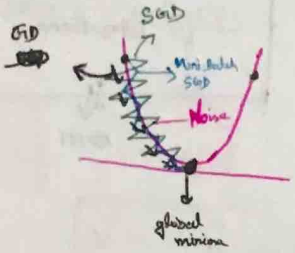
2nd " = 1000 "

3rd " = 1000 "

1000 " = 1000 record

Advantages:

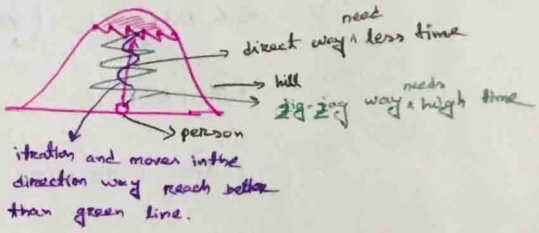
1. Doesn't resource intensive
2. Convergence will be better
3. Time complexity will reduce



SGD → jig-jag is very high. Noise also high

Mini Batch SGD → jig-jag is very low. Noise also minimal

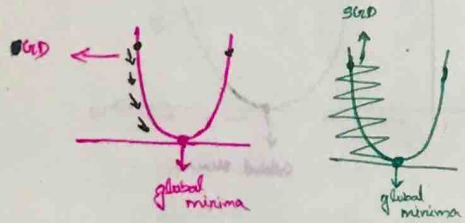
jig-jag is called noise

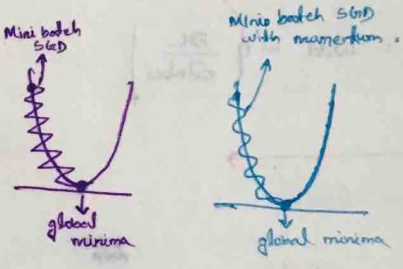


How do we remove Noise:

We use momentum.

Mini Batch SGD with momentum:





Exponential Weighted Average (moving)

⇓ used in

Time Series

⇓ model

ARMA , ARIMA

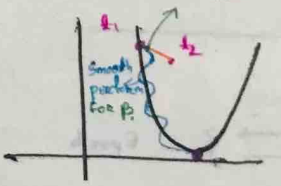
Weight Updation — $W_{new} = W_{old} - \eta \frac{\partial L}{\partial W_{old}}$

Bias " → $b_{new} = b_{old} - \eta \frac{\partial L}{\partial b_{old}}$

$$W_t = W_{t-1} - \eta \frac{\partial L}{\partial W_{t-1}}$$

current time, previous time

removing the noise & smoother the curve



Example:

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

$V_{t_1} = a_1$

$V_{t_2} = \beta V_{t_1} + (1-\beta) a_2$ (β: previous, (1−β): current)

$= (.95) V_{t_1} + (.05) a_2$

very important, Value is high — less important, Value is low

{ Forecasting }

β = hyper parameter (weight / exponential moving avg)

β = 0 to 1

⇓

.95

From β we can find importance ∧ previous or current timestamp value.

$V_{t_3} = \beta V_{t_2} + (1-\beta) a_3$

Exponential weighted/Moving Avg,

$$W_t = W_{t-1} - \eta V_{dw}$$

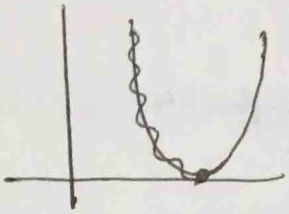
$$V_{dw_t} = \beta * V_{dw_{(t-1)}} + (1-\beta)\frac{\partial L}{\partial W_{t-1}}$$

Advantages:

(i) Reducing the noise

(ii) Mini Batch

(iii) Quicker convergence



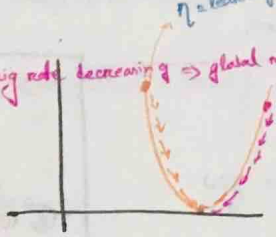
(v) Adagard (Adaptive Gradient Descent):

→ helps to change the learning rate

{fixed} = optimizer

η = learning rate, helps to speed the convergence.

η = fixed ⇒ Adaptive (changing η) ⇒ Learnig rate decreasing ⇒ global minima



$$W_t = W_{t-1} - \boxed{\eta \frac{\partial L}{\partial W_{t-1}}}$$

$$W_t = W_{t-1} - \eta' \frac{\partial L}{\partial W_{t-1}}$$

(Wnew, Wold)

decreasing η' = $\frac{\eta}{\sqrt{\alpha_t + \epsilon}}$ (η → small number (0.01))

epsilon, is a small number. if $t=0$, $\eta' = \infty$. to remove this problem ϵ is used.

$\eta \downarrow$ → decreasing ↓↓↓

When model was starting to train we should keep on decreasing it as we reach the global minima

Huge Number ← $\alpha_t = \sum_{i=1}^{t} \left(\frac{\partial L}{\partial W_t}\right)^2$ ↑↑↑ α_t increasing, can't control it

$t=1$, $\eta = 0.01$

$t=2$, $\eta = .05$

$t=3$, $\eta = 0.003$

when α_t is increasing, η' is decreasing.

because $\boxed{\eta' \propto \frac{1}{\sqrt{\alpha_t + \epsilon}}}$, Here $\frac{\eta}{\sqrt{\alpha_t + \epsilon}} = u$

$\therefore \eta'$ = small number. thats why $W_t \approx W_{t-1}$

(vi) Adadelta and RMSPROP: Exponential weighted average

$$\eta' = \frac{\eta}{\sqrt{S_{dw} + \epsilon}}$$ there will be a slow decreasing

$S_{dw_t} = 0$

$$S_{dw_t} = \beta S_{dw_{t-1}} + (1-\beta)\left(\frac{\partial L}{\partial W_{t-1}}\right)^2$$

$\beta = .95$

$$S_{dw_t} = (.95) S_{dw_{t-1}} + (.05)\left(\frac{\partial L}{\partial W_{t-1}}\right)^2$$

↑↑↑ increase small value. ... it can control

We can control S_{dwt} so that there will be a slow decreasing of this learning rate, η' and we will be able to reach the global minima.

(vii) Adam Optimizer: (best optimizer forever)

momentum + RMSPROP (Adaptive learning rate)

Combine with momentum along with the rmsprop instuition.

Instuition means, we have adaptive learning rate

diff types of Adam optimizer

(i) adam max

(i) Smoothing

(ii) Learning Rate Adaptive

$V_{dw} = 0$, $V_{db} = 0$, $S_{dw} = 0$, $S_{db} = 0$

Adam optimizer

$$W_t = W_{t-1} - \eta' V_{dw}$$

$$b_t = W_{bt-1} - \eta' V_{db}$$

$$\eta' = \frac{\eta}{\sqrt{S_{dw} + \epsilon}}$$

$$V_{dbt} = \beta * V_{db_{t-1}} + (1-\beta)\left(\frac{\partial L}{\partial V_{db_{t-1}}}\right)^2$$

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