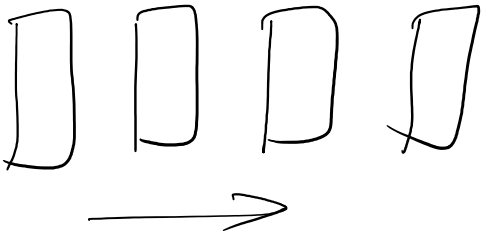
Introduction

05 July 2022 09:57

Performance of NN

↳ how the speed the training



{Optimizers} ✓

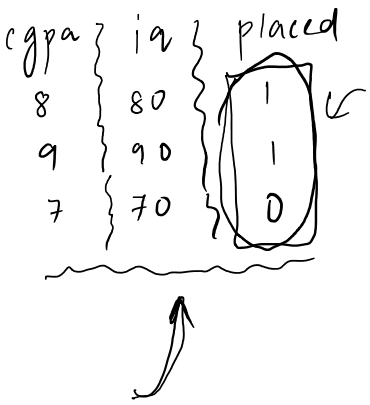
→ Weigh init

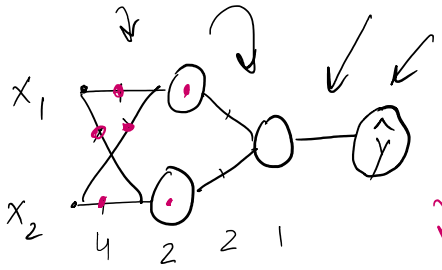
→ Batch Norm

→ Activation function

Role of Optimizer

05 July 2022 10:01





9 parameters

optimization

2 min

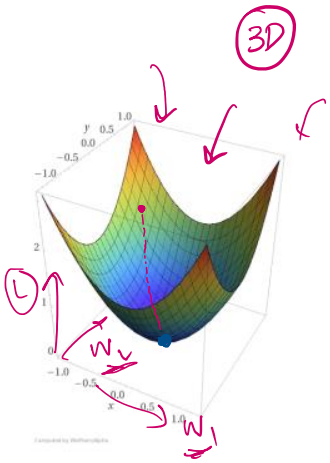
y $\hat{y}$ (w, b)

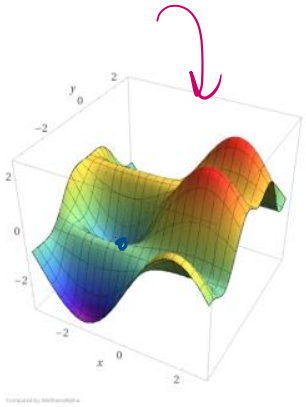
C/L()

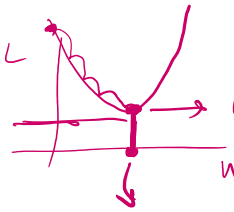
9 para

2 weights

3D $L(\_,\_,\_,\_,\_,\_)$ 10D







Gradient Descent

$$w_n = w_0 - \eta \frac{\partial L}{\partial w}$$

Types of Optimizers

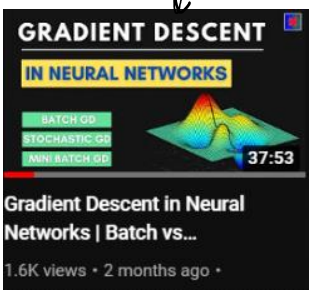
05 July 2022 10:02

- Batch GD
- Stochastic GD
- Mini batch GD

10

epochs = 10

$$W_n = W_0 - \eta \frac{\partial L}{\partial W}$$



Gradient Descent in Neural Networks | Batch vs...

1.6K views • 2 months ago •

500 rows → pred

↓

loss → weight update

10 × 500 → 5000

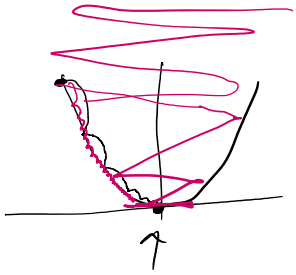
batch size = 100

5 batch → 10 × 5 times

Challenges

05 July 2022 10:02

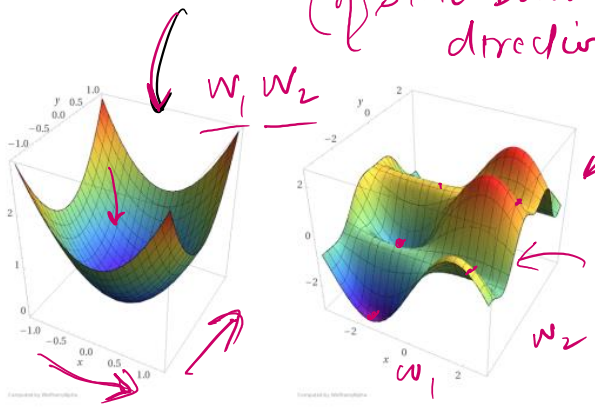
$$w_{n} = w_{0} - \eta \frac{\partial L}{\partial w_{0}}$$



1) learning rate

2) learning rate scheduling → pre define

3)



(η) same both the directions

(9) weights and bias 9 sep learn

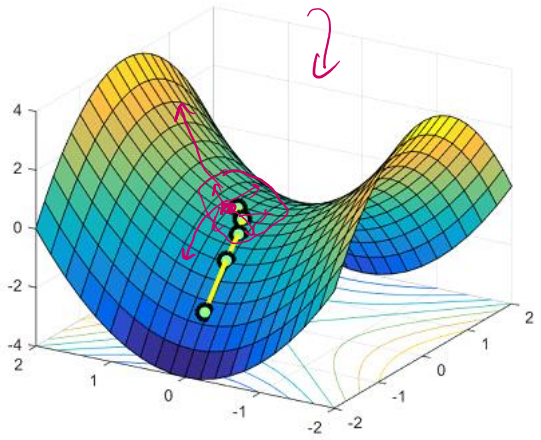
10-D → 10 directions

complex minima

sub optimum

4) local minima

5) saddle point

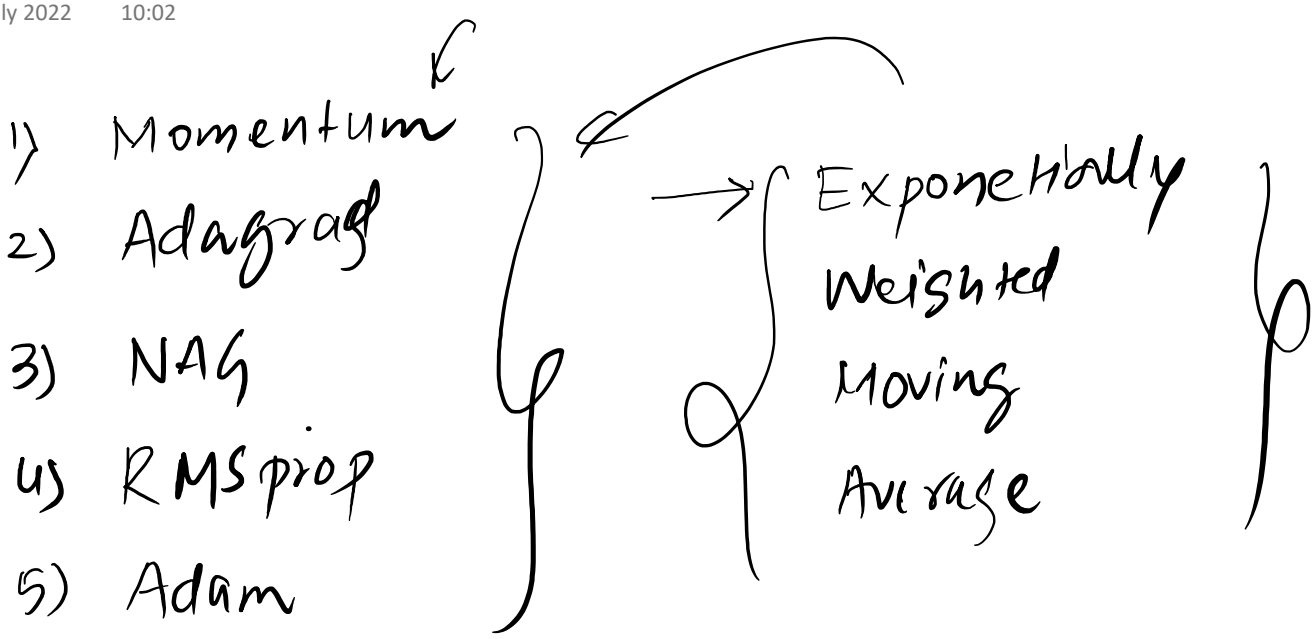


$$\frac{\partial L}{\partial w} = 0$$

$$w_{n} = w_{0} - \eta \frac{\partial L}{\partial w}$$

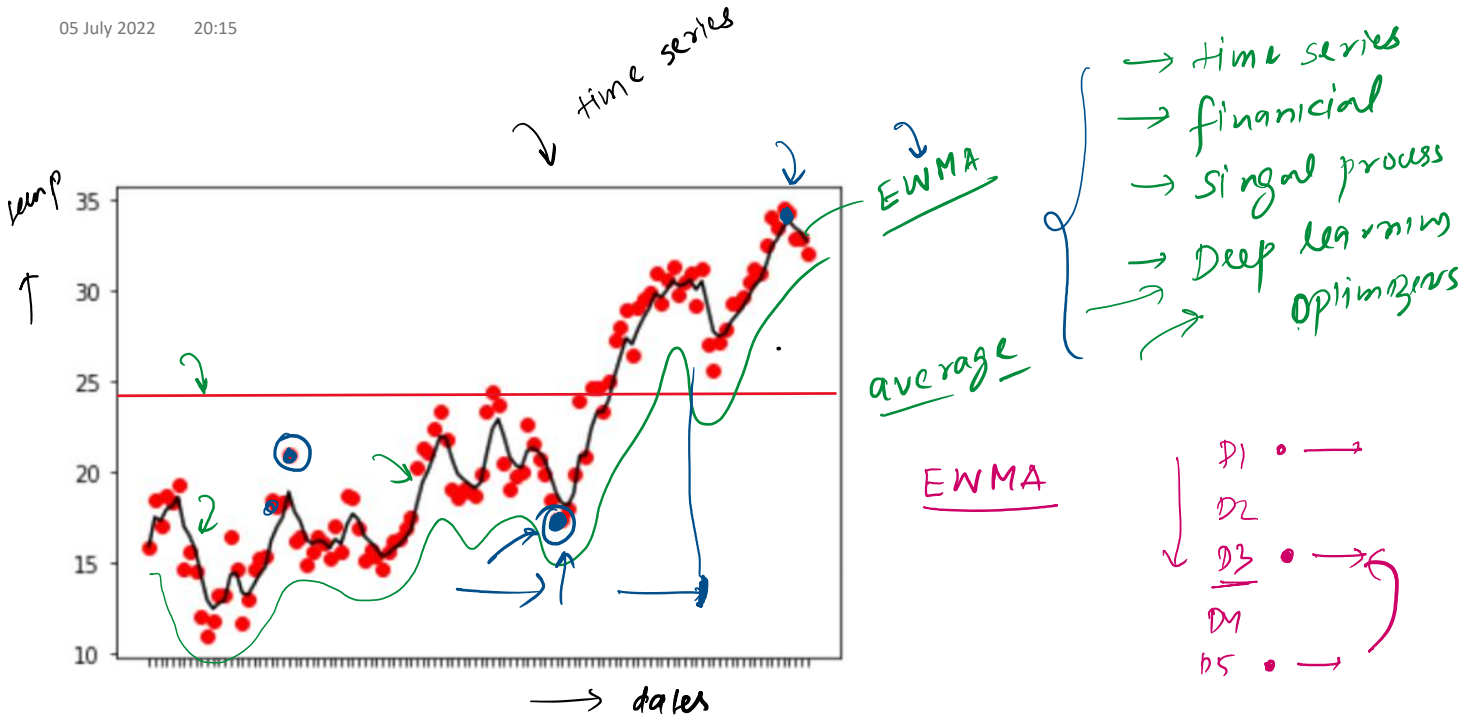
What next?

05 July 2022 10:02



The What

05 July 2022 20:15



Mathematical Formulation

05 July 2022 20:16

$$V_t = \beta V_{t-1} + (1-\beta)\theta_t$$

$\beta \rightarrow 0 < \beta < 1$

index	temp (θ)
D1	25
D2	13
D3	17
D4	31
D5	43

$\beta = 0.9$

$V_0 = 0$ $\quad V_0 = \theta_0$

$V_1 = 0.9 \times V_0 + 0.1 \times 13$

$= 1.3$

$V_2 = 0.9 \times 1.3 + 0.1 \times 17 =$

$\beta = 0.5$

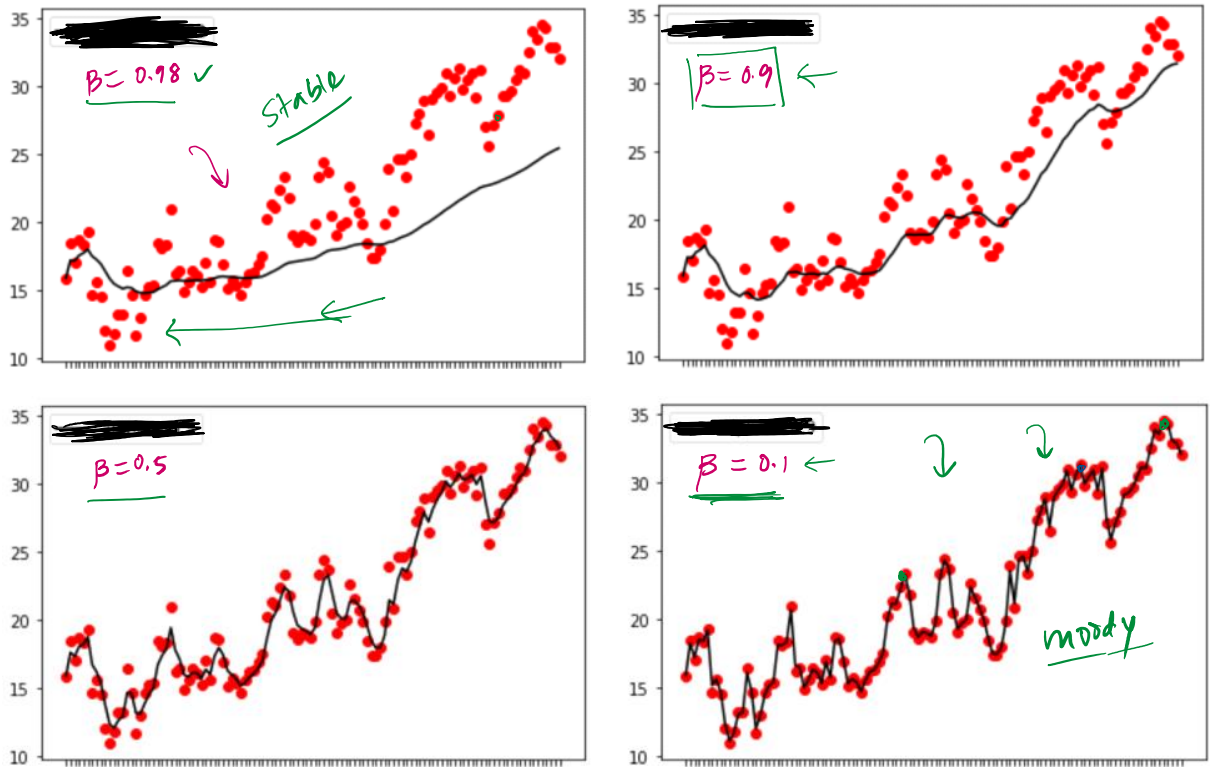
$\frac{1}{1-0.5} = \frac{1}{0.5}$

2 days

$\beta \rightarrow \boxed{\frac{1}{(1-\beta)}}$ $\rightarrow$ $\beta = 0.9$

$\frac{1}{1-0.9} = \frac{1}{0.1} = 10$

Effect of β



Mathematical Intuition

$$V_t = \beta V_{t-1} + (1-\beta)\theta_t$$

$$V_4 = \beta V_3 + (1-\beta)\theta_4$$

$$V_4 = \beta^3(1-\beta)\theta_1 + \beta^2(1-\beta)\theta_2 + \beta(1-\beta)\theta_3 + (1-\beta)\theta_4$$

$$V_t = \beta V_{t-1} + (1-\beta)\theta_t$$

$V_0 = 0$

$V_1 = (1-\beta)\theta_1$

$V_2 = \beta V_1 + (1-\beta)\theta_2$

$= \beta(1-\beta)\theta_1 + (1-\beta)\theta_2$

$V_3 = \beta V_2 + (1-\beta)\theta_3$

$= \beta^2(1-\beta)\theta_1 + \beta(1-\beta)\theta_2 + (1-\beta)\theta_3$

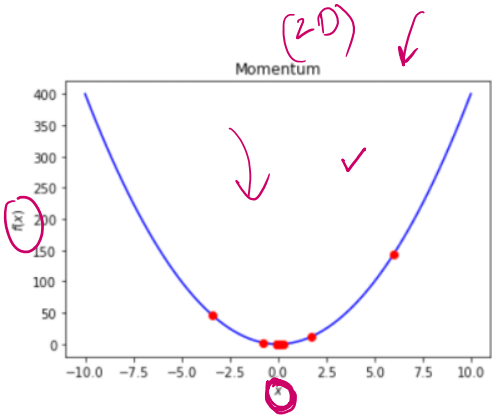
$V_4 = \beta^3(1-\beta)\theta_1 + \beta^2(1-\beta)\theta_2 + \beta(1-\beta)\theta_3 + (1-\beta)\theta_4$

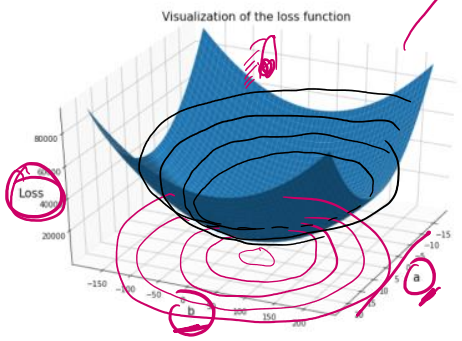
$= (1-\beta)[\beta^3\theta_1 + \beta^2\theta_2 + \beta\theta_3 + \theta_4]$

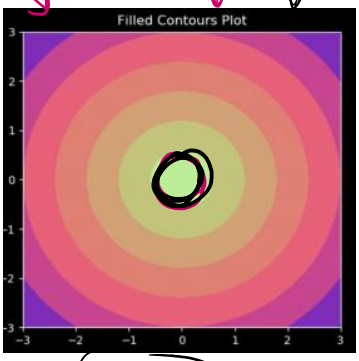
$\beta^3 < \beta^2 < \beta$ $\quad 0 < \beta < 1$

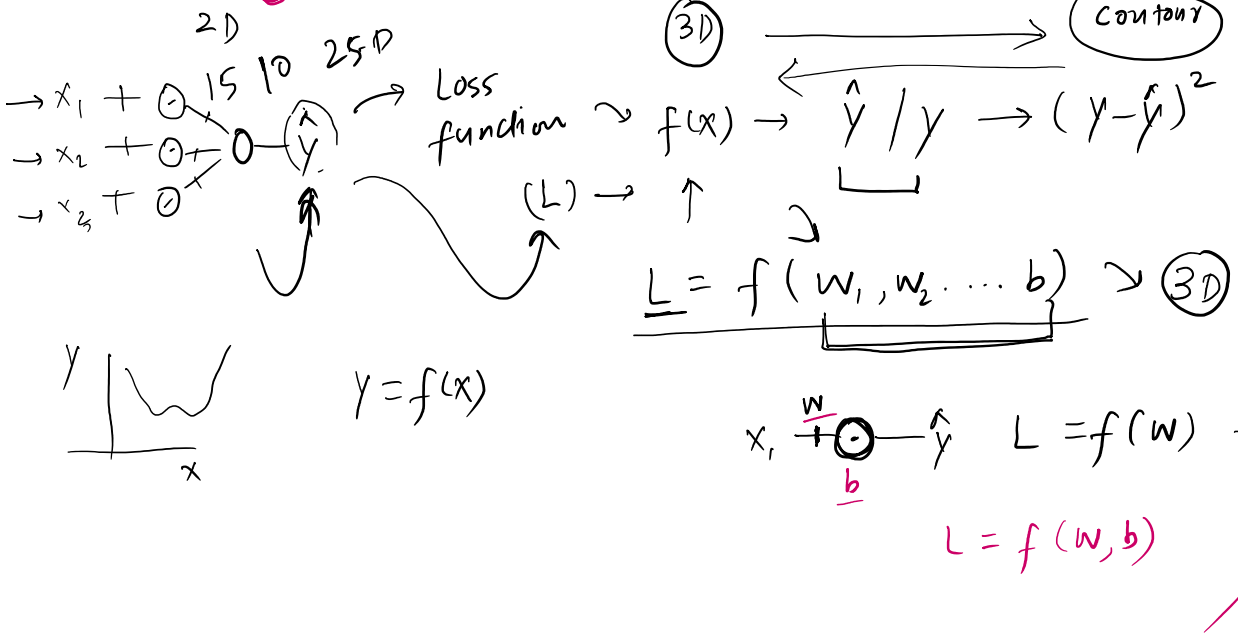
Understanding Graphs

20 July 2022 12:03



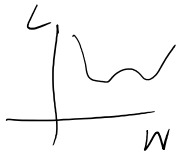






y = f(x)

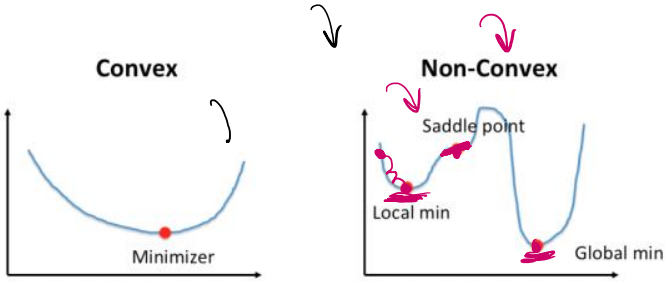
L = f(w)

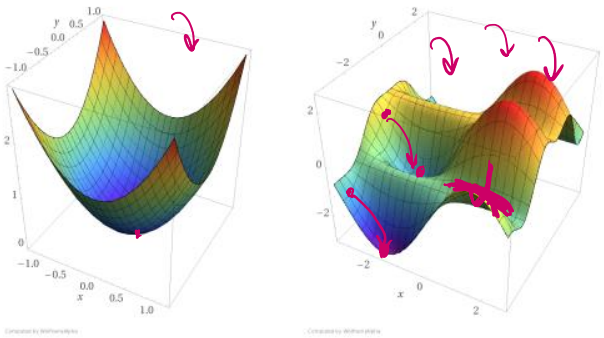


L = f(w, b)

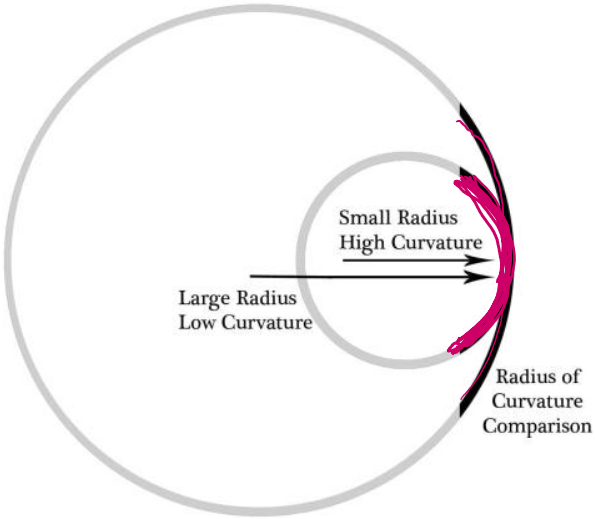
Convex Vs Non-Convex Optimization

20 July 2022 13:06



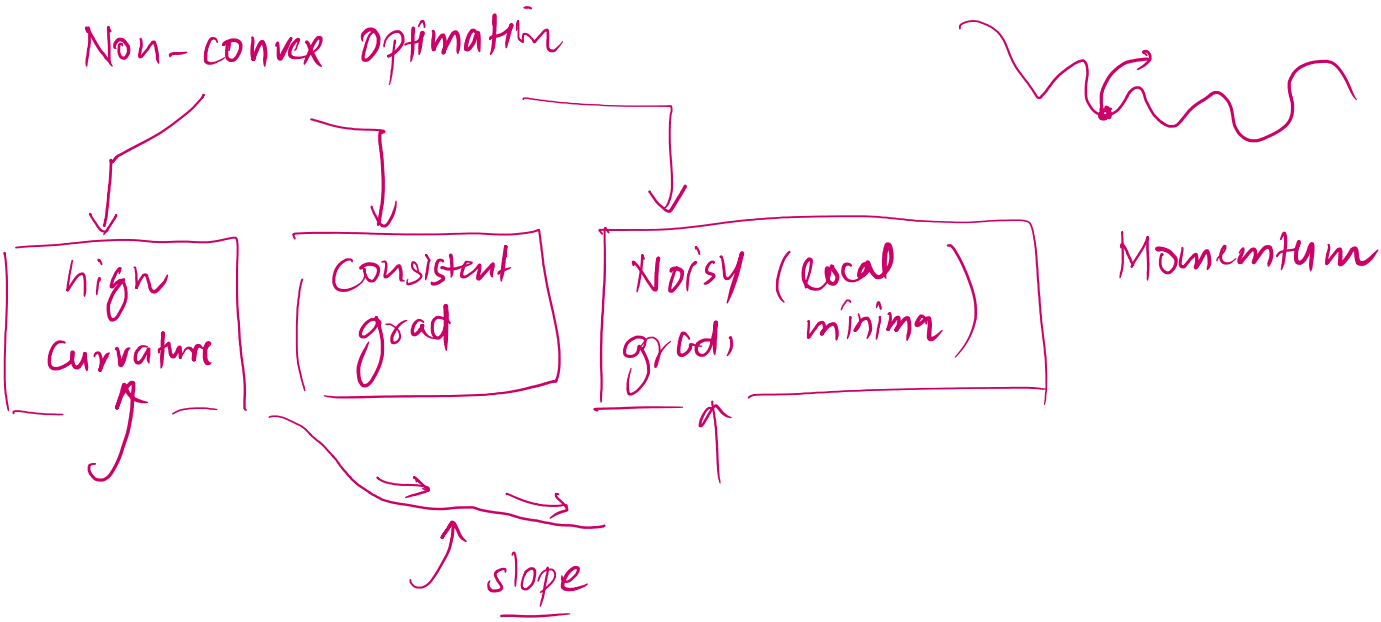


1) local minima
2) Saddle point
3) High curvature



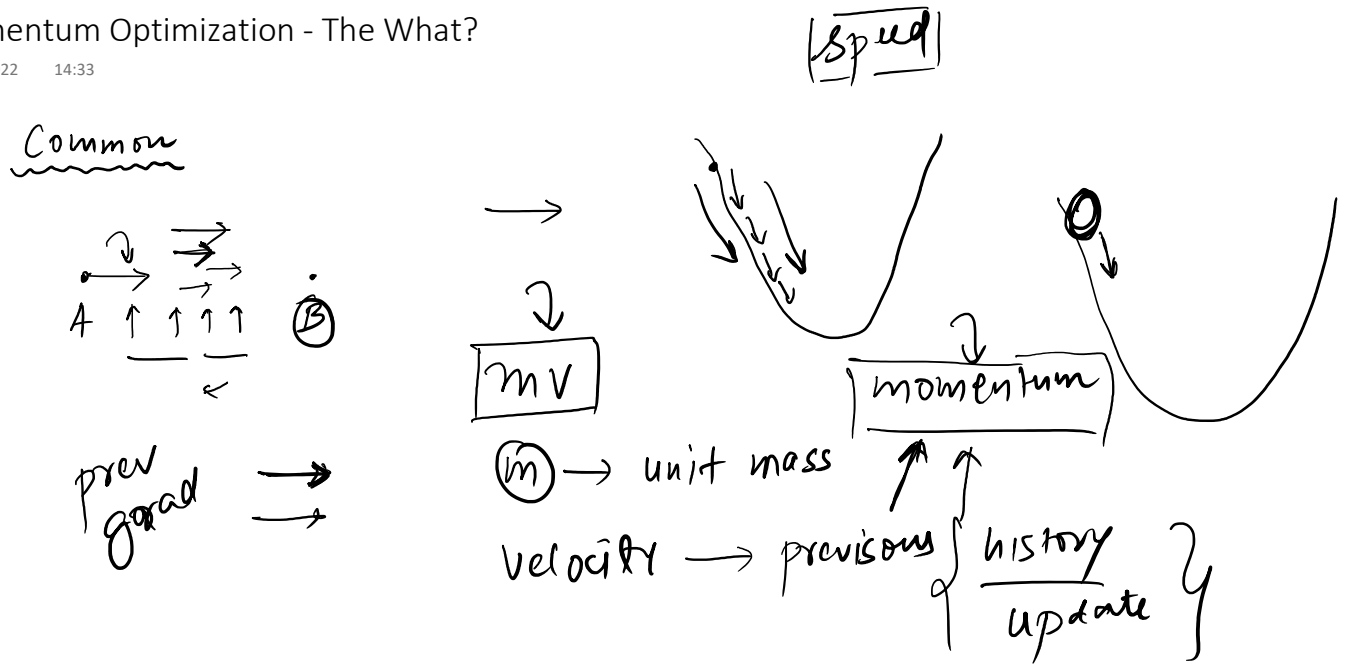
Momentum Optimization - The Why?

20 July 2022 14:28



Momentum Optimization - The What?

20 July 2022 14:33



Momentum Optimization - Mathematics(The How?)

20 July 2022 14:56

$$w_{t+1} = w_t - \eta \nabla w_t$$

$v \rightarrow$ velocity

$$w_{t+1} = w_t - v_t$$

$$v_t = \beta * v_{t-1} + \eta \nabla w_t$$

acceleration

momentum

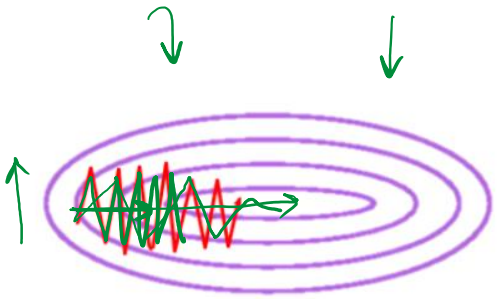
0.9

$0 < \beta < 1$

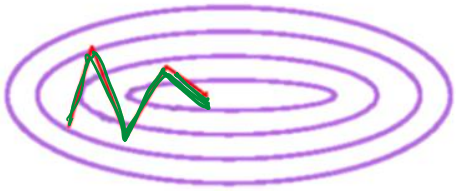
history of past velo- use prev velocity

v_{t-2}

v_{t-3}



SGD



SGD momentum

faster convergence

Effect of beta

20 July 2022 15:09

$$w_{t+1} = w_t - v_t$$

$$v_t = \beta v_{t-1} + \eta \nabla w_t$$

$$w_{t+1} = w_t - \eta \nabla w_t$$

SGD

$\beta = 0$ momk → SGD

$\beta = 0$

decaying factor

$\frac{1}{1-0.9} = \frac{1}{0.1} = 10$

$\frac{1}{1-\beta}$

0.9

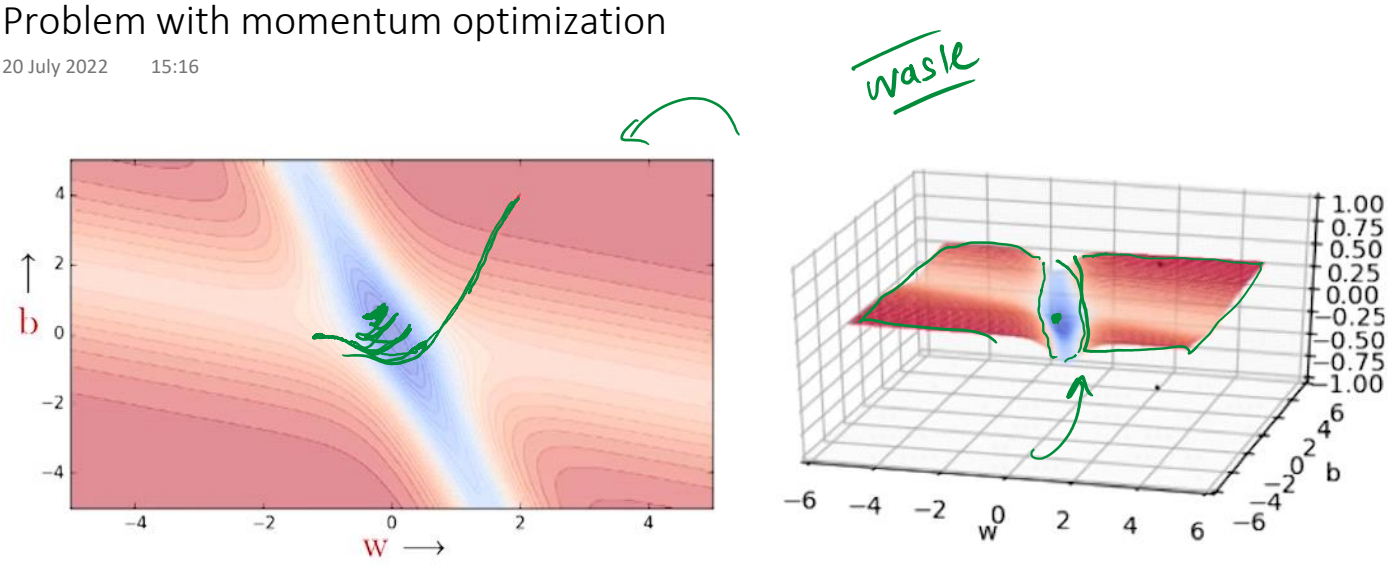
current avg past 10 veloci

up1 up2 . . up3

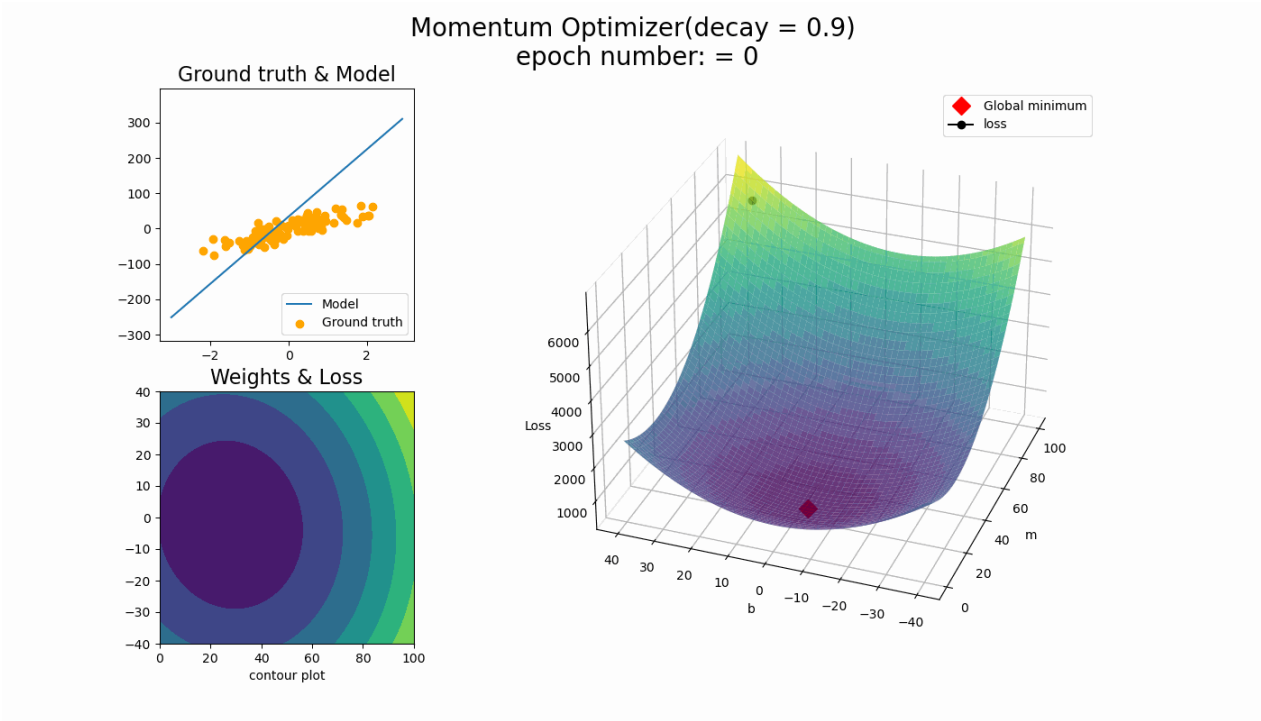
$\beta = 1$

Problem with momentum optimization

20 July 2022 15:16

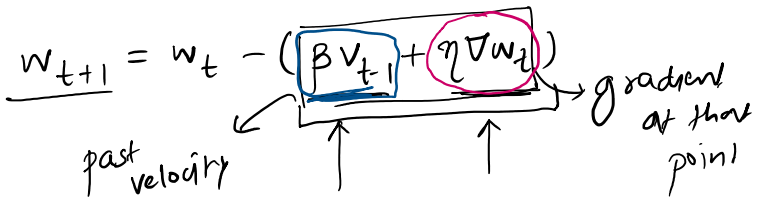


decay 0.9



Mathematical Intuition

24 July 2022 13:17



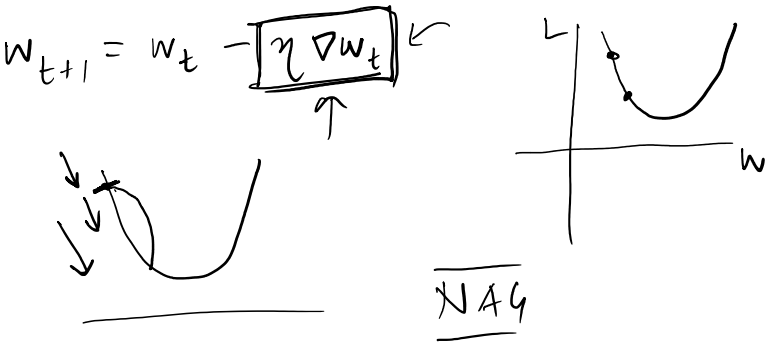
Momentum

$w_{t+1} = w_t - v_t$

where

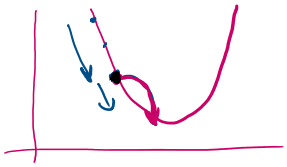
$v_t = \beta v_{t-1} + \eta \nabla w_t$

$\beta \rightarrow$ decay factor

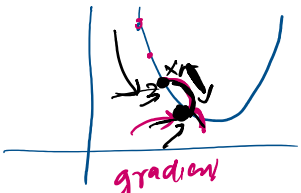


NAG

Nesterov Accelarated Gradient better



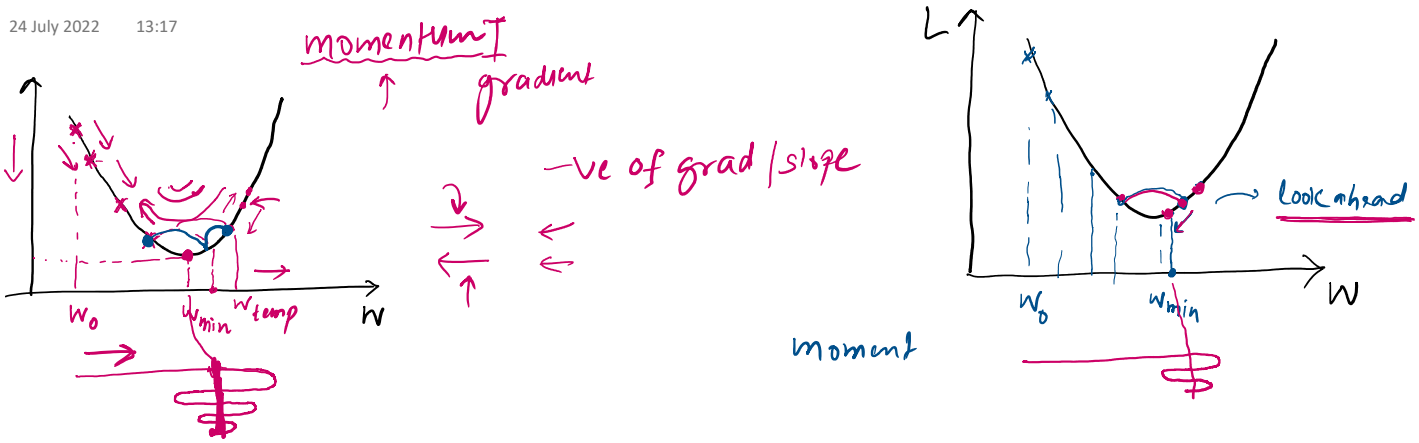
history of velocity + gradient at that point



$$\begin{cases} w_{la} = w_t - \beta v_{t-1} \\ v_t = \beta v_{t-1} + \eta \nabla w_{la} \rightarrow v_t = (w_t - w_{la}) + \eta \nabla w_{la} \\ w_{t+1} = w_t - v_t \end{cases}$$

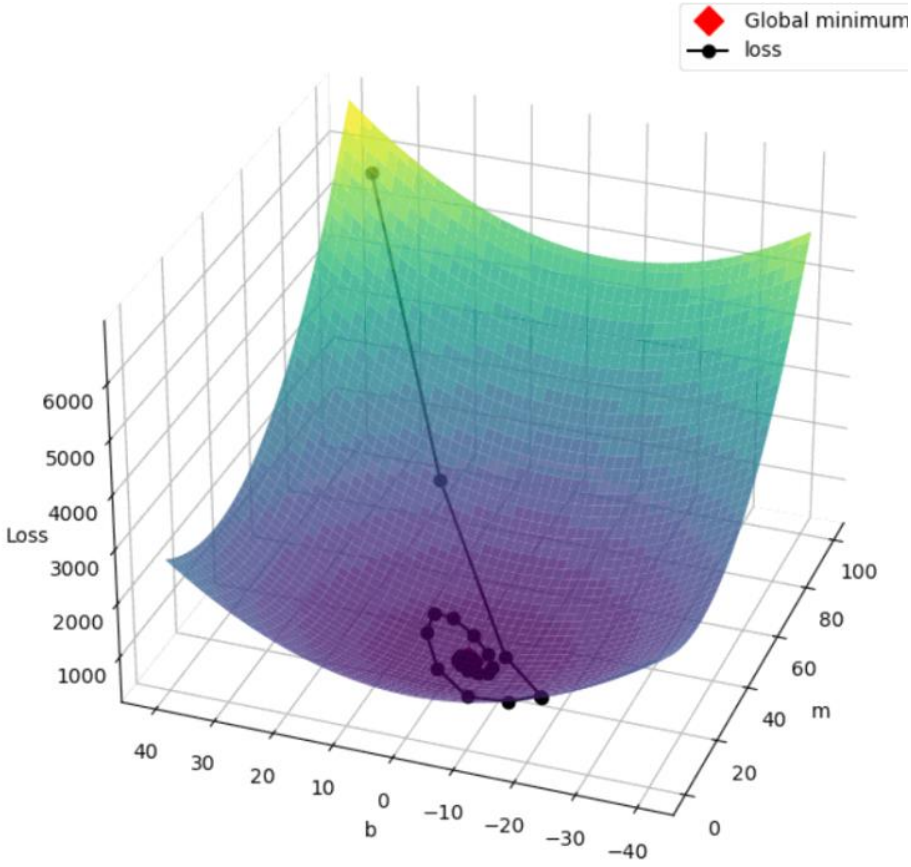
Geometric Intuition

24 July 2022 13:17



Disadvantage

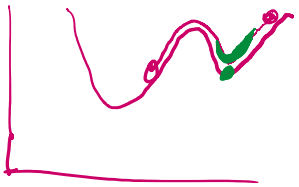
24 July 2022 13:18



Disadvantage

NAG → oscillation → dampen

local minima



Keras Code

24 July 2022 13:18

```
tf.keras.optimizers.SGD(
    learning_rate=0.01, momentum=0.0, nesterov=False, name="SGD", **kwargs
)
```

SGD

Momentum
momentum = 0.9
nesterov = False

NAG
momentum = 0.9
nesterov = True

AdaGrad Intro

02 August 2022 18:45

Adaptive Gradient → learning rate fix X change adapt →

→ input features → scale ↳ different ↘

⇒ features → [Sparse] → 0 (most)

↳ Adagrad

cgpa | salary | bank loan

0-10 ↓ 0 – Lacs

↓ Normalize

iq | cgpa | [from iit] | package

0 → ↘ 0,0,0,

1 →

Elongated bowl problem

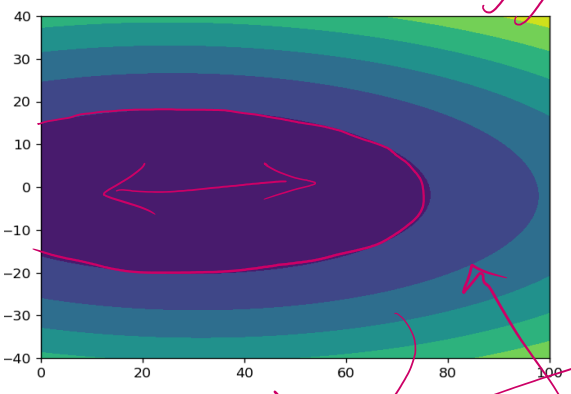
sparse

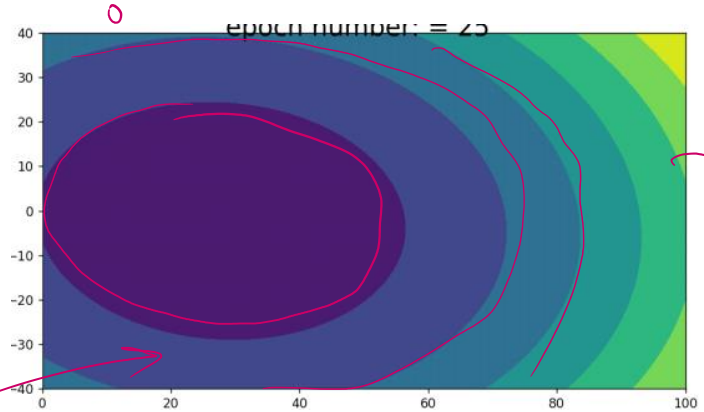
x | y

0

0

0

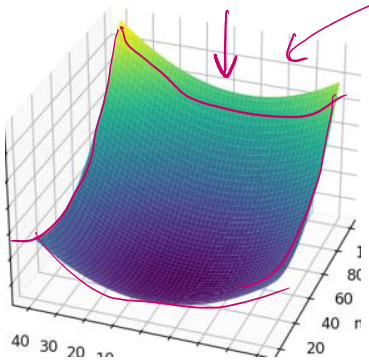


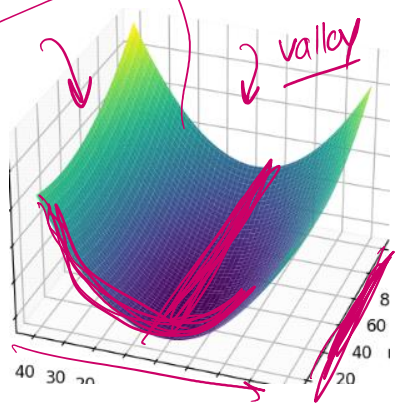


→ m

m

loss → w,b → w,b





valley

Bgd → momentum → NAG

Adagrad

elongated

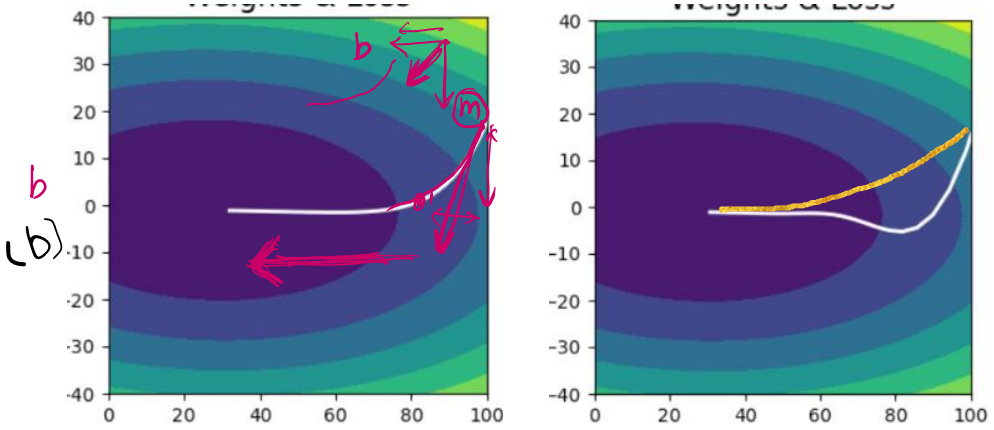
↓

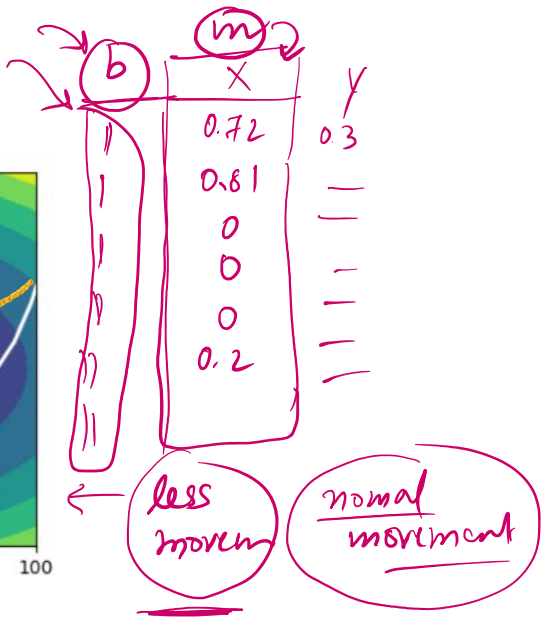
slope → 1axis – chang

2nd axis constant

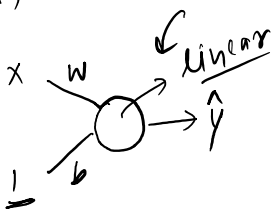
How optimizers behave(Why?)

02 August 2022 18:44





$(y-\hat{y})^2$



BGD → sparspe

for i in epochs:

$$w = w - \eta\left[\frac{\partial L}{\partial w}\right]$$

$$b = b - \eta\left[\frac{\partial L}{\partial b}\right]$$

→ big update in every epoch

$$\left[\frac{\partial L}{\partial w}\right] = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial w}$$

$$= -2(y-\hat{y})[x] \quad \text{add}$$

$$\frac{\partial L}{\partial b} = [-2(y-\hat{y})*1] \quad \text{bias num}$$

100 rows → 0 → X sparse

small number

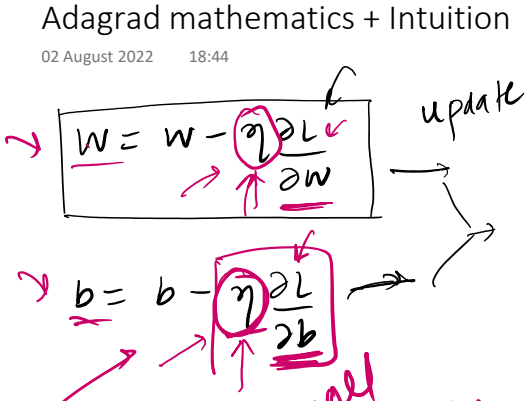
small in every epoch

sparse col → small

non sparse → normal / big

Adagrad mathematics + Intuition

02 August 2022 18:44



Adagrad

optimization

lr → same

big ↘ lr → small update - small

Adagrad → diff learning rates

past gradients sum(squared)

lr → small → gradient → big

epoch

$V_t = V_{t-1} + (\nabla W_t)^2$

V_t $\left(\frac{\partial L}{\partial W}\right)^2 \left(\frac{\partial L}{\partial b}\right)^2$

$\frac{\partial L}{\partial W}$

$W_{t+1} = W_t - \eta \nabla W_t$

$W_{t+1} = W_t - \eta \Delta W_t$

η small

$\sqrt{V_t + \epsilon}$ η ϵ = small num

$V_t = 0$

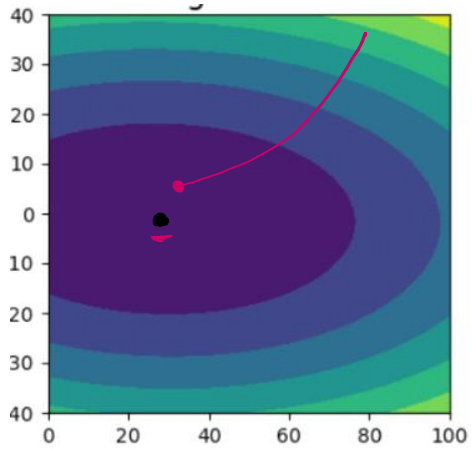
(-ve +ve)

sparse data

We do square as we want mangnitude , not do mod as we have to do backpropgation also.

Disadvantage

02 August 2022 18:43



.

Initially it is fast to reach near , but later this update become small , and very hard to converge towards global minima

NN → use X ↘ linear regress

η → small + decrease

$\sqrt{v_t}$

v_t

past updates ↳ gradient

epochs

update small

global minima

↘ RMSprop
↘ Adam

The Why?

03 August 2022 14:03

RMSprop → Root Mean square Prop

↳ improvement

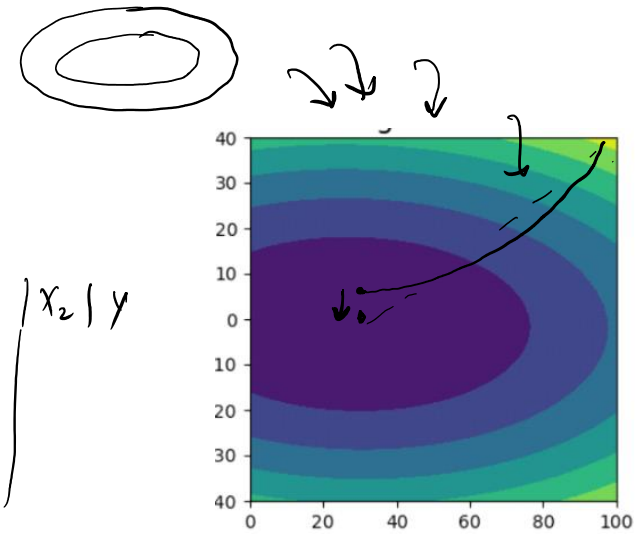
Adagrad ← sparse data

BGD → momentum

Disadvantage → learning rate

x_1	x_2	y
0		
0		
0		
1		
0		

Update → small



Mathematical Formulation

03 August 2022 14:04

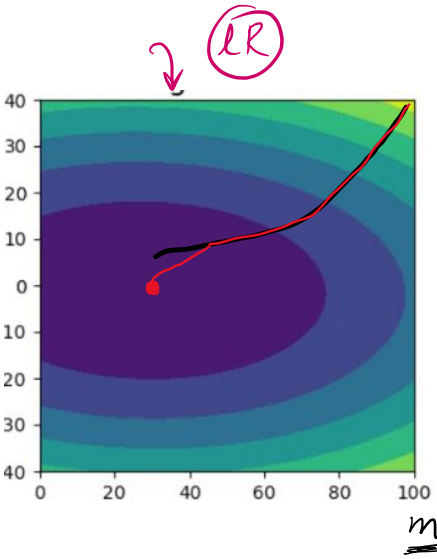
epochs

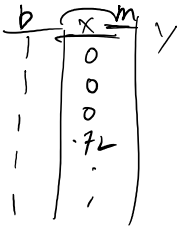
Adagrad

$v_t = v_{t-1} + (\nabla w_t)^2$ — gradients

$w_{t+1} = w_t - \frac{\eta}{\sqrt{v_t + \epsilon}} \nabla w_t$

(b) small





↓ b move

non-sparse

RMSprop

$v_t = \beta v_{t-1} + (1-\beta)(\nabla w_t)^2$ — exp decaying avg

$\beta = 0.95$

old epoch

$w_{t+1} = w_t - \frac{\eta}{\sqrt{v_t + \epsilon}} \nabla w_t$

→ $v_0 = 0$

→ $v_1 = 0.95 \times 0 + 0.05 (\nabla w_1)^2$

→ $v_2 = 0.95 \times 0.05 (\nabla w_1)^2 + 0.05 (\nabla w_2)^2$

→ $v_3 = 0.95 \times 0.95 \times 0.05 (\nabla w_1)^2 + 0.95 \times 0.05 (\nabla w_2)^2 + 0.05 (\nabla w_3)^2$

largest

epoch 3

epoch 2

epoch 1

smaller

v_t shoot → η small

Not Performs good in non-convex optimization.

Adagrad → NN → non-convex optimization

convex optimiz → linear reg

Performs good in convex optimization.

↘ convex → linear
optimiz reg

global minima RMSprop

Disadvantage

03 August 2022 14:54

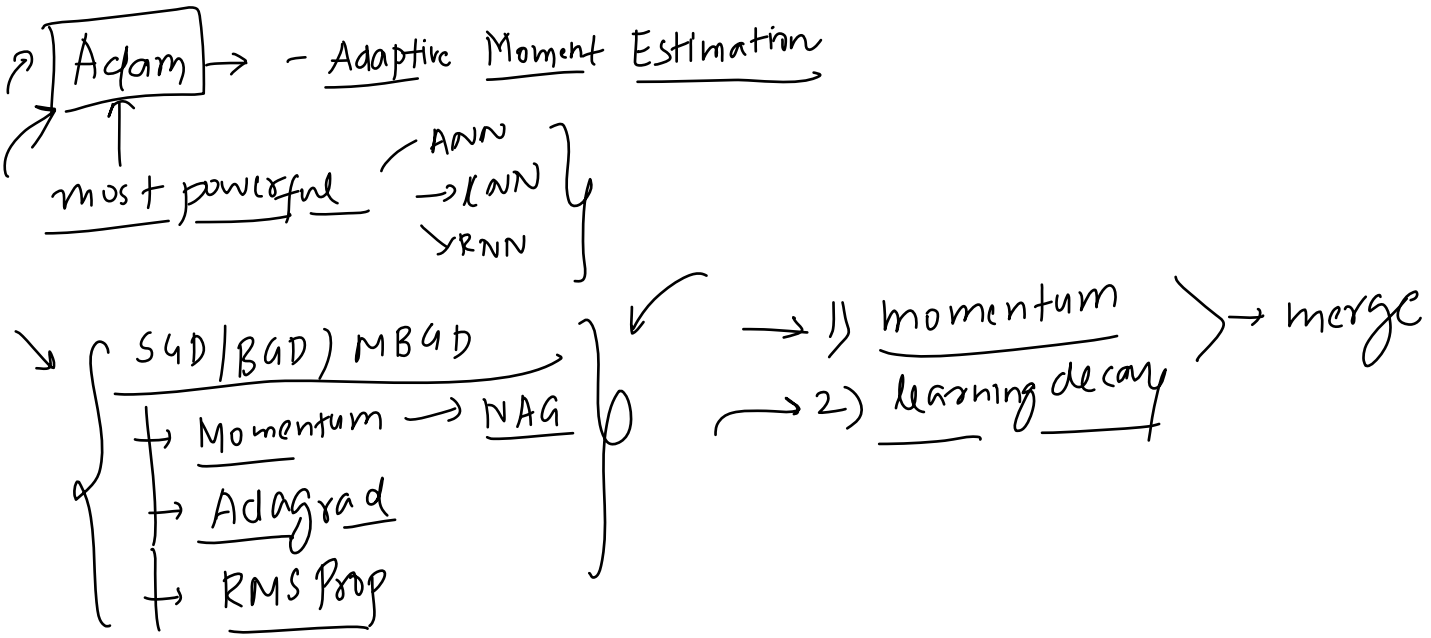
Before adam is there , we used RMSProp only .
Infact today if adam not works , we consider RMSProp

No,
Optimization

↓
Adam

Introduction

04 August 2022 10:45



Momentum : It introduces the concept of momentum i.e using history values also. (ewma on gradients)

NAG : It uses same momentum concept with little modification of the point where gradients are calculated.

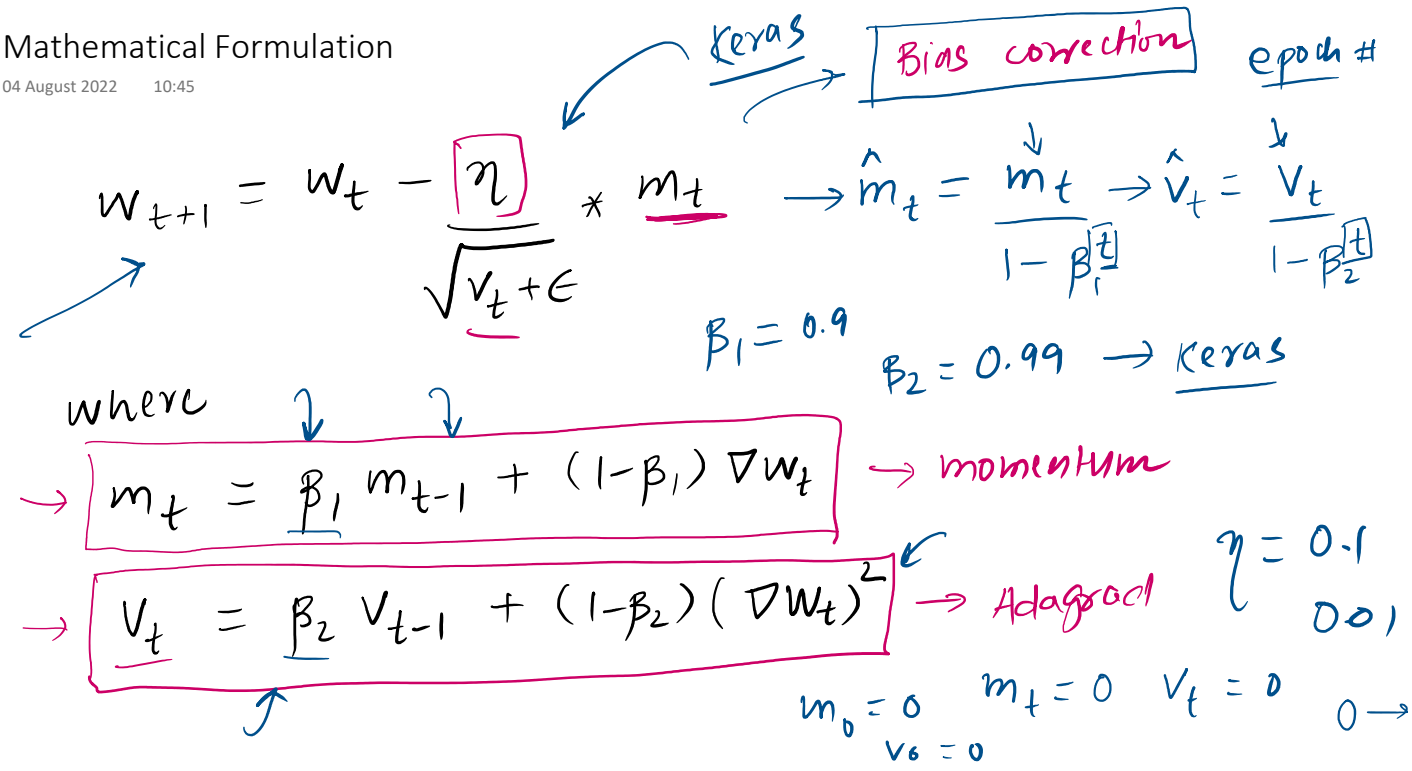
Adagrad : It considers change learning rate.

RMSProp : In overtime , adagrad fails to converge towards global minima due to very low learning rate, it adds exp weighted moving avg to modify learning rate.

Adam : It merges concept of both momentum and learning rate. By using concept of momentum for gradients and learning rate change .

Mathematical Formulation

04 August 2022 10:45



This bias correction is there , as in initial weights as zero so kick off we used it.

Verdict

04 August 2022 10:45