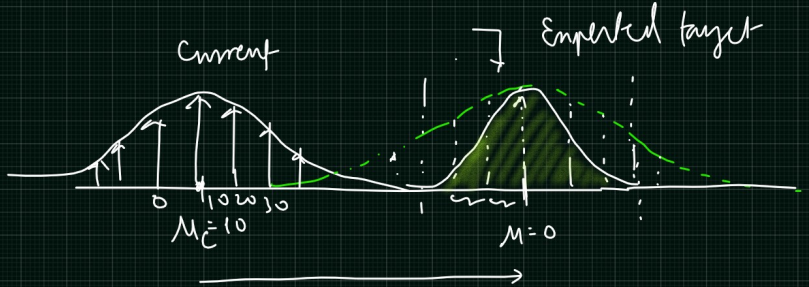
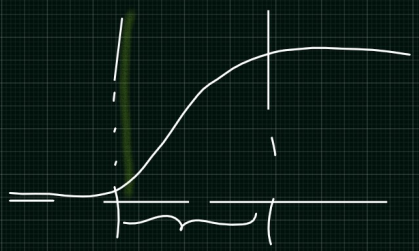
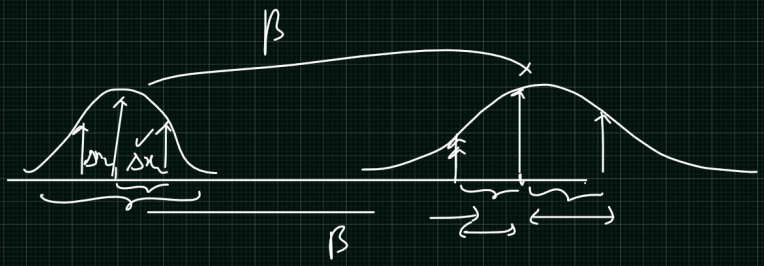
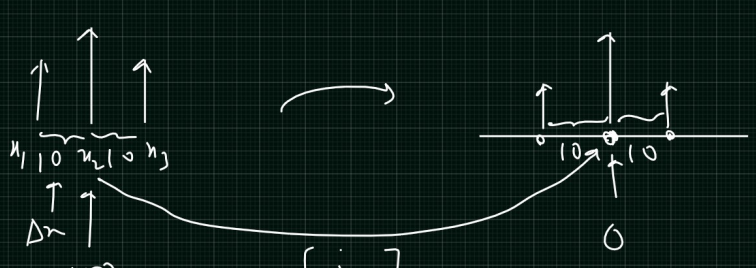
4) $z^{(i)} = \gamma \otimes \hat{x}^{(i)} + \beta$

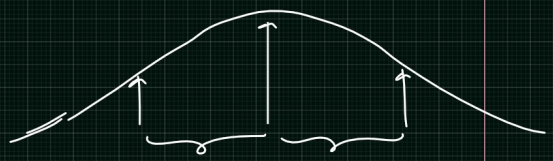
scaling [ ] shifting





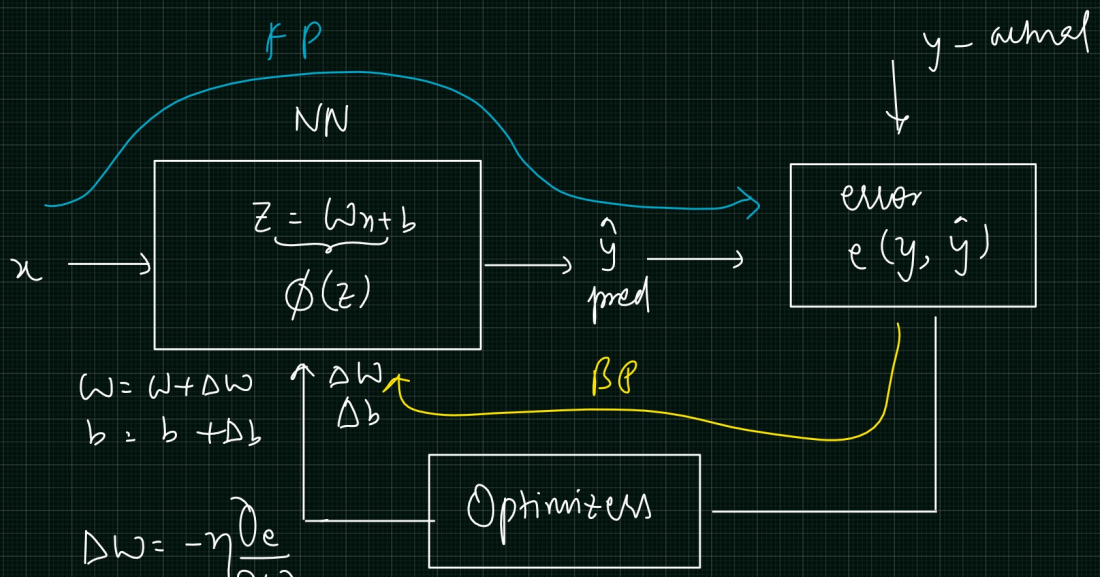






$2 \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + 100$

FAST OPTIMIZERS



Gradient descent

Observation on Grad. descent

$$w_{new} = w_{old} - \eta \left.\frac{\partial e}{\partial w}\right|_{w = w_{old}}$$

$\frac{\partial e}{\partial w}(w = w_{old})$

or

$$\theta = \theta - \eta \nabla_\theta J(\theta)$$

$J(\theta)$ → cost function

θ → parameters (w, b)

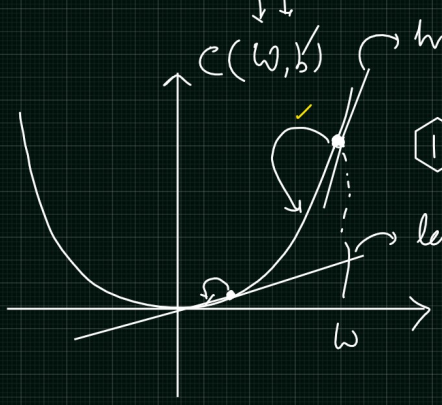
$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ b_1 \\ b_2 \\ \vdots \end{bmatrix}_{\theta} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ b_1 \\ b_2 \\ \vdots \end{bmatrix}_{\theta} - \eta \underbrace{\begin{bmatrix} \partial J/\partial w_1 \\ \partial J/\partial w_2 \\ \vdots \end{bmatrix}}_{\nabla_\theta J(\theta)}$$

$$w = w - \eta \frac{\partial C}{\partial w}$$

$$\Delta w = -\eta \frac{\partial C}{\partial w} = (-\eta)\frac{\partial C}{\partial w}$$

$\Delta w \propto \frac{\partial C}{\partial w}$ → slope of the tangent

$\Delta \Theta \propto \nabla_\Theta J(\Theta)$ → cost fn.

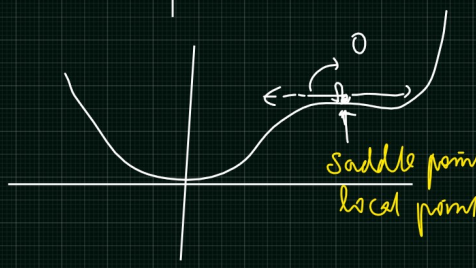


$\Delta w \uparrow$

$\Delta w \downarrow$

if $\frac{\partial C}{\partial w} \uparrow$ ⇒ slope steep

if $\frac{\partial C}{\partial w} \downarrow$ ⇒ near flat surface ⇒ flat surface



② Δw only depends on current weight

$C(y, \hat{y})$ $\quad \frac{\partial C}{\partial w}$

$\uparrow$

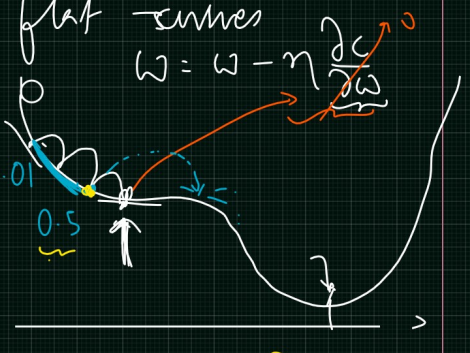
$C(y, \sigma(wx+b))$

⇒ it doesn't get influenced by previous wt-updates.

③ Grad. descent is very slow at near flat curves

⇒ slow training

because GD. doesn't considers previous updates



$\Delta w \propto \{-\eta\} \frac{\partial C}{\partial w}$

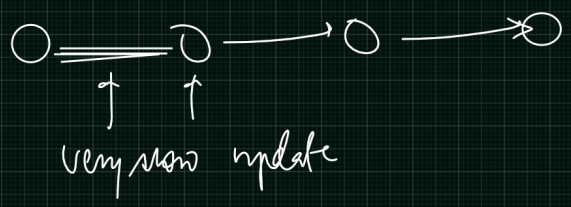
$0.5 - 0.3 =$

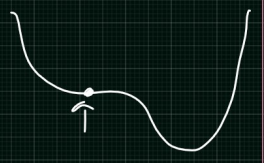
100 ⟶ 0.03 $\quad$ 70%

⑤ Learning rate scheduler : { callbacks }

1^{st} η_1, 2^{nd} η_2

⑥ Chain rule → $\frac{\partial e}{\partial w} = \frac{\partial e}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial w}$





Sol: Fast Optimizers

i) Momentum Optimization ✓
ii) Nesterov Accelerated Gradient (NAG)
iii) AdaGrad ✓
iv) RMSProp ✓
v) Adam Optimizers ✓

Momentum Optimization :-

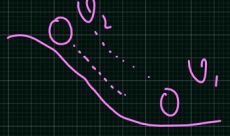
Given by Boris Polyak in 1964

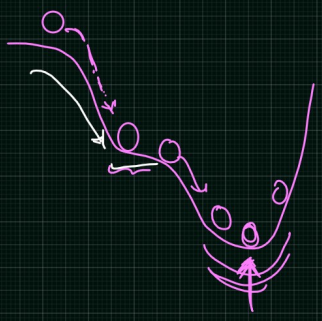
Idea :-

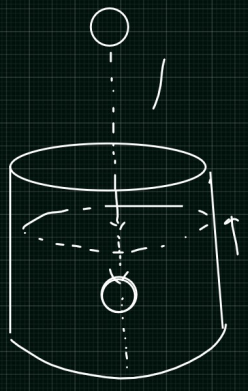
Motivation

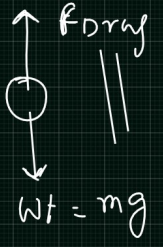
$p = mv$

$F = \frac{dp}{dt} = m\frac{dv}{dt} = ma$





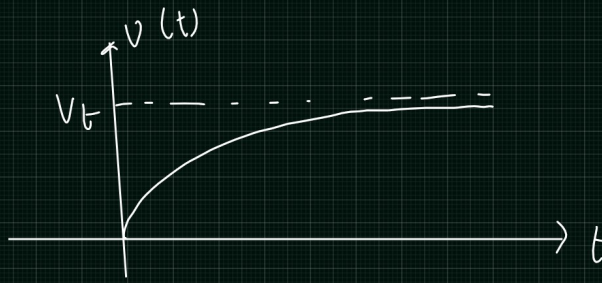




$F_{DRAG} = Wt \Rightarrow a = 0$

v = const

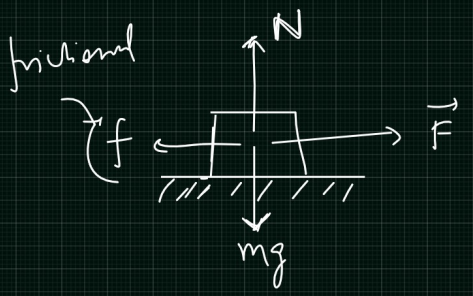
$v_{terminal}$ velocity



Algorithm of momentum optimization

m = momentum

β = coeff of momentum
analogous to
coefficient of friction



$f = \mu N$

↓

coeff of friction

$\mu = 0$ ⇒ smooth surface $\quad f = 0$

Algo.

$$m \leftarrow \beta m + \eta \nabla_\theta J(\theta)$$

$$\theta \leftarrow \theta - m$$

or

$$m = \beta m + \eta \frac{\partial C}{\partial w}\Big|_{w = w_{current}}$$

$$w = w - m$$

assumption
bias = 0

$$\theta = \theta - \left[\beta m + \eta \nabla_\theta J(\theta)\right]$$

if $\beta = 0$

$$\theta = \theta - \beta m - \eta \nabla_\theta J(\theta) \qquad w = w - \beta m - \eta \frac{\partial C}{\partial w}$$

$$\boxed{\theta = \theta - \eta \nabla_\theta J(\theta)} \quad \text{GD} \qquad w = w - \eta \frac{\partial C}{\partial w}$$

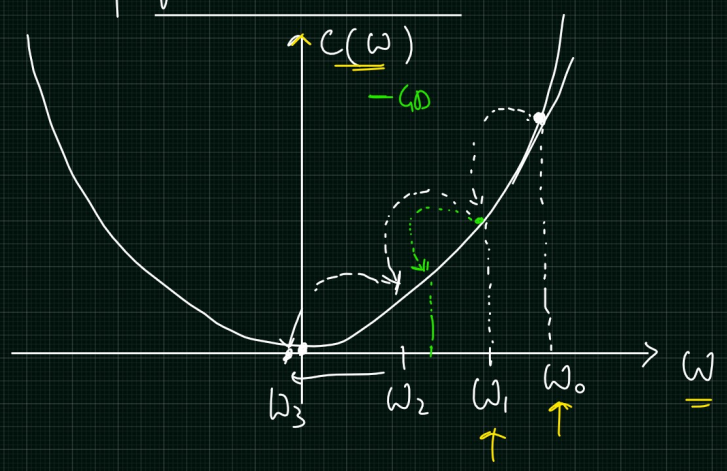
GD

→ Through experimentation it has observed that

$\beta = 0.9$ works well

but you are allowed to tweak

Graphical Observation :-



$C \to C(y, \hat{y}) \to C(\eta, \tilde{w}, b)$

Step 1

initial, $w = w_0$, $m_0 = 0$, $\beta = 0.9$

$$m_1 = \beta m_0 + \eta \left.\frac{\partial C}{\partial w}\right|_{w=w_0}$$

(m_0 → 0, $\frac{\partial C}{\partial w}$ → +ve)

$$m_1 = \eta \left.\frac{\partial C}{\partial w}\right|_{w=w_0} \quad -①$$

$$w_1 = w_0 - m_1$$

$$\boxed{w_1 = w_0 - \eta \left.\frac{\partial C}{\partial w}\right|_{w=w_0}} \quad -②$$

(+ve, +ve)

Equivalent to GD

Step 2 :

$w = w_1$, $\beta = 0.9$, $m_1 = \eta \left.\frac{\partial C}{\partial w}\right|_{w=w_0}$

$$m_2 = \beta m_1 + \eta \left.\frac{\partial C}{\partial w}\right|_{w=w_1}$$

$$m_2 = \beta \eta \left.\frac{\partial C}{\partial w}\right|_{w=w_0} + \eta \left.\frac{\partial C}{\partial w}\right|_{w=w_1}$$

$$= \eta \left[\beta \left.\frac{\partial C}{\partial w}\right|_{w=w_0} + 1 \times \left.\frac{\partial C}{\partial w}\right|_{w=w_1} \right]$$

0.9 past — 90% η past

current — 100% η present

$$\to w_2 = w_1 - m_2$$

$$\{ w_2 = w_1 - \eta \underbrace{\left[\beta \left.\frac{\partial C}{\partial w}\right|_{w=w_0} + \left.\frac{\partial C}{\partial w}\right|_{w=w_1} \right]}_{\Delta w_{m_0}} \quad m_0 \text{ eqn}$$

$$\{ w_2 = w_1 - \eta \left.\frac{\partial C}{\partial w}\right|_{w=w_1} \} \; GD$$

$$\Delta w_{GD}$$

$$\Delta w_{mo} > \Delta w_{GD}$$

Step 2:

$$m_2 = \eta \left[\beta \left.\frac{\partial C}{\partial w}\right|_{w=w_0} + \left.\frac{\partial C}{\partial w}\right|_{w=w_1} \right]$$

$w = w_2$, $\beta = 0.9$

$$m_3 = \beta m_2 + \eta \left.\frac{\partial C}{\partial w}\right|_{w=w_2}$$

$$m_3 = \beta \eta \left[\beta \left.\frac{\partial C}{\partial w}\right|_{w=w_0} + \left.\frac{\partial C}{\partial w}\right|_{w=w_1} \right] + \eta \left.\frac{\partial C}{\partial w}\right|_{w=w_2}$$

$$= \eta \left[\beta^2 \left.\frac{\partial C}{\partial w}\right|_{w=w_0} + \beta \left.\frac{\partial C}{\partial w}\right|_{w=w_1} + \left.\frac{\partial C}{\partial w}\right|_{w=w_2} \right]$$

0.9×0.9

0.81

81% of past-

90% of recent past

100% Amount

more accumulation

$$w_3 = w_2 - m_3$$

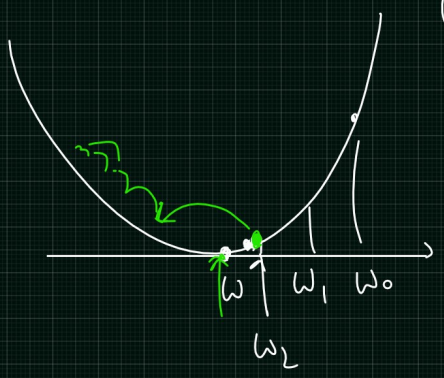
$$m_3 = \eta\left[\beta^2 \frac{\partial C}{\partial w}\Big|_{w=w_0} + \beta \frac{\partial C}{\partial w}\Big|_{w=w_1} + \frac{\partial C}{\partial w}\Big|_{w=w_2}\right]$$

$\beta^2 \frac{\partial C}{\partial w}\Big|_{w=w_0} + \beta \frac{\partial C}{\partial w}\Big|_{w=w_1}$: past $\neq 0$ ✓

$\frac{\partial C}{\partial w}\Big|_{w=w_2}$: current $\to 0$

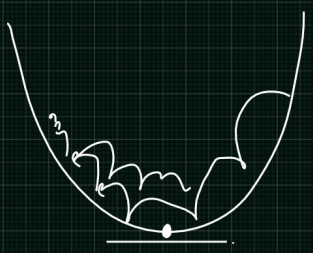
$m_3 \neq 0$

$$w_3 = w_2 - m_3$$



Drawbacks ✓

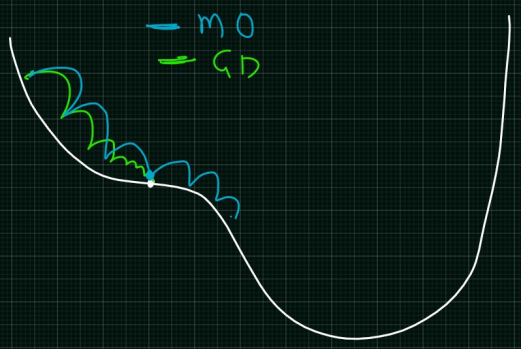
① It oscillates when it reaches closer to minima (local / global) because of accumulation of past gradients



β^2 β^3 β^4 ...

0.81 0.81×0.9 ↓ ....

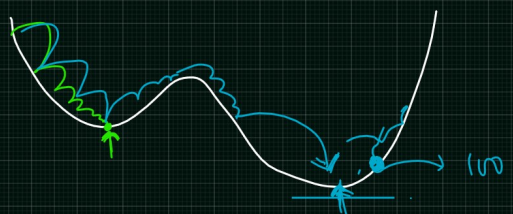
$0.9 \times 0.9 = 0.81$ $0.9 > 0.81$



Advantages

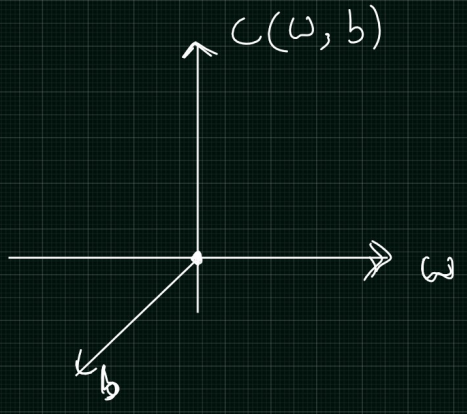
↳ Momentum can help in fast convergence

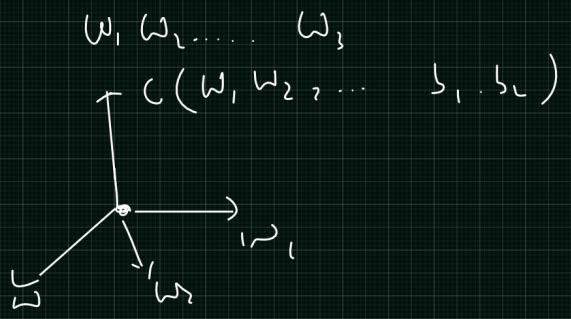
↳ Oscillation / accumulation can help in overcoming local minima

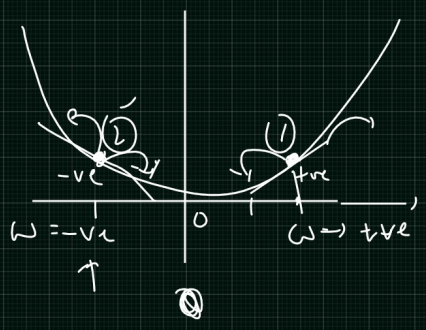


Keras

⇒ optim = tf.keras.optimizers.SGD(lr = 0.01, momentum = 0.9)
↑
β





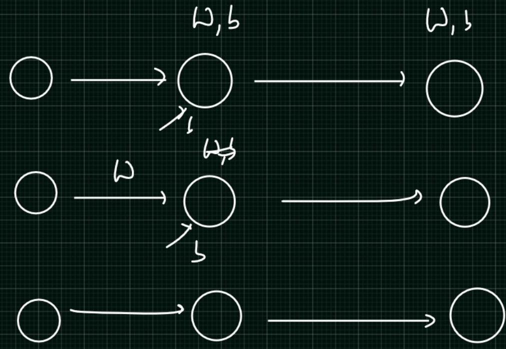


$$w = w + \Delta w$$
$$= w - \eta \frac{\partial C}{\partial w}\Big|_{w = w_{current}}$$

① $w - (+ve)(+ve)$
$+ve - +ve =$

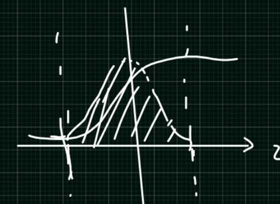
② $w - (+ve)(-ve)$
$-ve + (+ve)$

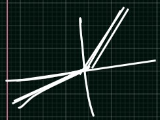
$100 - 15 = 85$



$z = Wx + b$

$\sigma(z)$





PReLU

α

Hardware

accuracy

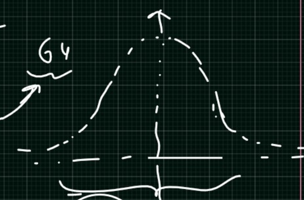
float

Traffic...

float16

speed, accuracy

float32 → G4



BN

β, γ