16 July Day 4

24 November 2022 06:52

Stochastic gradient Descent with Momentum.

GD → NG → Curve smooth Diff

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

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

1000 [100 → ; 10 → no. ba] 50 — MB

B1 B2 - B3 . . . 15th 20th 21 (WB) — 10, 7 ↑ WB

—xox—

$v_{dwt} = \beta \times v_{dwt-1} + (1-\beta) \times \frac{\partial L}{\partial w_{t-1}}$

1st previous day / Today.

β = importance 0.8 = 5th, 0.9 = 10th

1000

① 2 3 4 5 6 7 8 9 10

WB Low, Low, Low — WB high → Noise → EMA

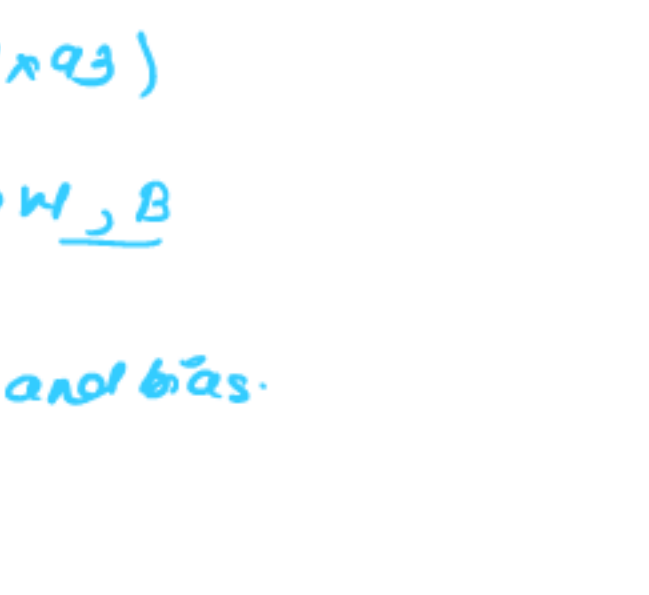
$V_{dw} = 0$
$V_{db} = 0$

$w_{new} = w_{old} - \alpha \frac{\partial L}{\partial w_{old}}$ → derivative of Loss w.r.t weights.

$v_{dwt} = v_{dwt-1} \times \beta + (1-\beta) \times \frac{\partial L}{\partial v_{dwt}}$

t1	t2	t3	t4	---	tn
a1	a2	a3	a4	---	an

$v_{t1} = \beta \times PD + (1-\beta) \times a_1$ (PD = 0) 0.9
$= 0.1 \times a_1$



$v_{t2} = \beta \times v_{t1} + (1-\beta) \times a_2$
$= 0.9 \times (0.1 \times a_1) + (1-\beta) \times a_2$
$= 0.9 \times (0.1 a_1) + 0.1 \times a_2$

$v_{t3} = \beta \times v_{t2} + (1-\beta) \times a_3$
$= 0.9 \times (0.9 \times 0.1 a_1) + 0.1 \times a_2) + (0.1 \times a_3)$

P10 W, B

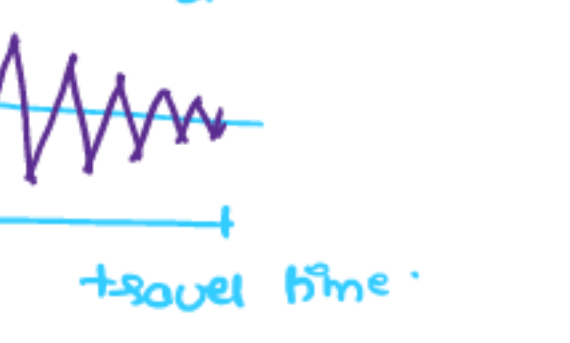
β = 0.8 5th

✓✓ ✓ ✓ ✓ 1 P1 P1 → current weight and bias. → new.

noise: SGD; ↓ noise: MBSGD; ↓ noise: SGDWM

β = 0 the algorithm become MBSGD

β = 1 algorithm goes into dynamic equilibrium keep moving on error surface.

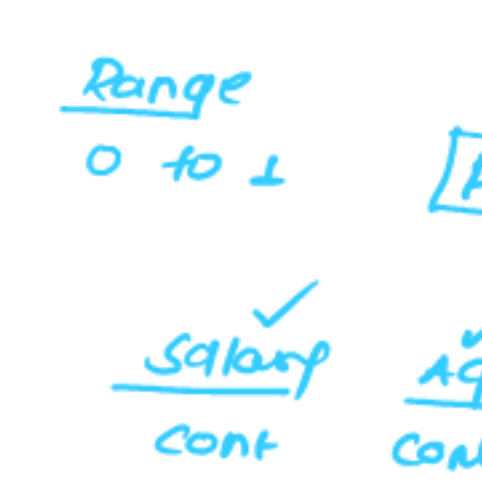


By using Exponential moving avg. we have avg out the oscillations in vertical direction close to zero.

It enables our algorithm to take straight path in forward direction to global minima and dampen but the vertical oscillation.

β value ↑ smoothening ↑
β value ↓ smoothening ↓

Adagrad



GD ∝ constant
SGD ∝ constant
MBSGD ∝ constant
GDWM ∝ constant

Core Idea → **Dynamic Learning Rate.**

Range 0 to 1 — $p = 0.01$

Salary ✓	Age ✓	gender ✓
cont	cont	M/F

In Real world model will learn dense feature 1st and then sparse feature.

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

$w_{new} = w_{old} - \alpha' \frac{\partial L}{\partial w_{old}}$ → Dynamic LR

$\alpha' = \frac{\alpha}{\sqrt{n_t + \epsilon}}$ — α' → new LR; α → old LR; ϵ → small positive no.; n_t → Derivative of loss w.r.t weights

ε A small positive number to avoid zero division Error.
$= 10^{-7}, 10^{-8}$ ✓

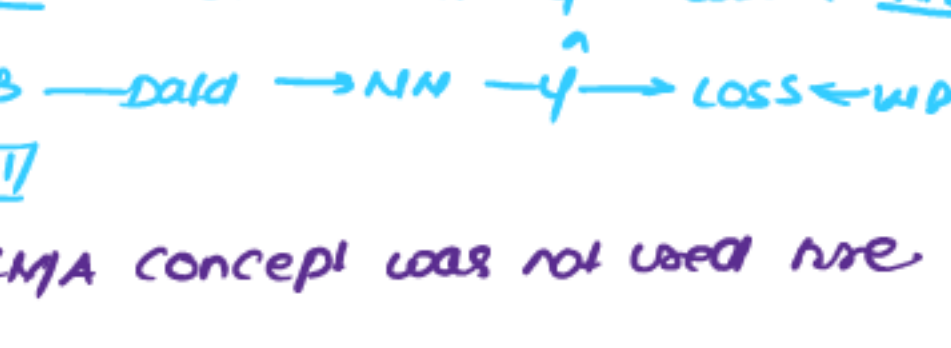
$n_t = \sum_{i=1}^{t} \left(\frac{\partial L}{\partial w_t} \right)^2$ — Summation.

B1 - - - - B5 - - - B10 ; B50 ; 49

$n_t \propto \frac{1}{\alpha'}$

Batch	n_t	α_t
1	↑	↑ ↑ ↑ ↑
2	↑ ↑	↑ ↑ ↑
3	↑ ↑ ↑	↑ ↑
4	↑ ↑ ↑ ↑	↑
	Increase	Decrease.

0.01, 0.009, 0.007, 0.004, 0.001 ↓



Problem faced here is called Learning Rate Decay.

Data → NN → $\hat{y}$ → Loss ← WB → WB
WB → Data → NN — $\hat{y}$ — Loss ← WB (+sum)
WB — Data → NN — $\hat{y}$ → Loss ← WB (α') — W SN, BN
→ α'

EMA concept was not used here

① Learning Rate reduces monotonically.

Ada delta or RMS prop.

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

$w_{new} = w_{old} - \alpha' \frac{\partial L}{\partial w_{old}}$ → Dynamic Learning Rate.

$\alpha' = \frac{\alpha}{\sqrt{S_{dw} + \epsilon}}$ → so now they have introduced the concept of EMA again to get rid of Learning Rate decay.

$S_{dw} = \beta \times S_{dwt-1} + (1-\beta) \left(\frac{\partial L}{\partial w} \right)^2$ — $\beta = 0.8$

It will not allow decay of Learning Rate to that extent. $\beta = 0.9$

1B — 10Bth → 50Bth
Adag 1 — 50th (W) $\left(\frac{\partial L}{\partial w}\right)$ to $\sum_{i=1}$
Adadelta (β) → 0.8 → 5 Day

44 45 46 47 48 [49] 50th
↑ ↑ ↑ ↑ ↑ ↑

* Adadelta is nothing but extension of Adagrad and also trying to solve the problem Learning Rate Decay.

(imp) In Adadelt we will not try to accumulate all past sq. gradient values instead tries to Restric the window of accumulated past gradient

---