SGDWM

15 September 2023 07:07

Stochastic gradient descent with momentum

$\beta = 0.9$ — [Batch]

$w_{new} = w_{old} - \alpha \frac{\partial L}{\partial w_{old}}$ } Deri of Loss w.r.t deri of weight

$b_{new} = b_{old} - \alpha \frac{\partial L}{\partial b_{old}}$ } Deri of Loss w.r.t deri of bias.

$w_{new} = w_{old} - \alpha\, v_{dw}$ → velocity component.

Random

Data → NN → $\hat{y}$ → Loss ← WB

$D = \beta \times$ previous Days $+ (1-\beta) \times$ Today
(β → 0.8)
$= \beta \times PD + (1-\beta)T.$

$v_{dw_t} = \beta \times v_{dw_{t-1}} + (1-\beta) \times v_{dw_t}$.

(Previous weights) (Current weight)

t_1 t_2 t_3 t_4 ... t_n

(x) a_1 a_2 a_3 a_4 ... a_n.

Random

$\beta = 0.8$

$v_{t1} = \beta \times PD + (1-\beta) \times T.$

$= 0.8 \times 0 + (1-0.8) \times a_1$

$= 0.2 a_1$ ✓ ①

—xox—

$v_{t2} = \beta \times PD + (1-\beta) \times T.$

$= \beta \times (0.2 a_1) + (0.2)\, a_2$

$= 0.2 a_1 \beta + 0.2 a_2$ ✓ ②
(β → 0.8)

$v_{t3} = \beta \times PD + (1-\beta) \times T.$

$\beta \times (0.2 A_1 \beta + 0.2 A_2) + 0.2 \times a_3$

$=$

[$\beta = 0.8$] → ⑤

50th. Last
49
48
47
46
45

By using exponential moving avg we tend out to avg oscillation in vertical direction close to zero.

β [↑] smoothing [↑]
β [↓] smoothing [↓.]

It enables the Algorithm to take straight path in forward direction to global minima and dampen out vertical oscillation.

Adagrad.

GD α constant
SGD α constant
MBSGD α constant
GDWM α constant.

Case Ldeq.
→ dynamic learning Rate

$\alpha = 0$ to 1

$\alpha = 0$ model is not learning anything

$\alpha = 1$ overshooting

$\alpha = 0.01$ preferred

$w_{new} = w_{old} - \alpha \frac{\partial L}{\partial w_{old}}$ (→ 0.)

$w_{new} = w_{old} - 0 \times$

$w_{new} = w_{old}$.

age	salary	gender.
0 -120	0 — ∞	M/F
Dense ✓	Dense ✓	Sparse ✓

In real world datasets model learn 1st dense feature and then sparse feature

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

—xox—xox—

$w_{new} = w_{old} - \alpha' \frac{\partial L}{\partial w_{old}}$. (→ dynamic LR) ✓

$\alpha' = \frac{\alpha}{\sqrt{n_t + \epsilon}}$ (New LR; α → old LR; ε → epsilon $10^{-7}, 10^{-\infty}$)

[ε] small positive no. used to avoid zero Division Errors.

[n_t] $n_t = \sum_{i=1}^{t} \left(\frac{\partial L}{\partial w}\right)^2$ ✓

Random

0.01 0.009

Data → NN → $\hat{y}$ → Loss ← WB, α

wβα.
Data → NN → $\hat{y}$ → Loss ← wβα.

$\alpha' = \frac{\alpha}{\sqrt{n_t + \epsilon}}$

[$\alpha' \propto \frac{1}{\sqrt{n_t}}$]

① $\sum_{i=1}^{1} \left(\frac{\partial L}{\partial w}\right)^2$ 5

② $\sum_{i=1}^{2} \left(\frac{\partial L}{\partial w}\right)^2_1 + \left(\frac{\partial L}{\partial w}\right)^2_2$ 10

③ $\sum_{i=1}^{3} \left(\frac{\partial L}{\partial w}\right)^2_1 + \left(\frac{\partial L}{\partial w}\right)^2_2 + \left(\frac{\partial L}{\partial w}\right)^2_3$ 15

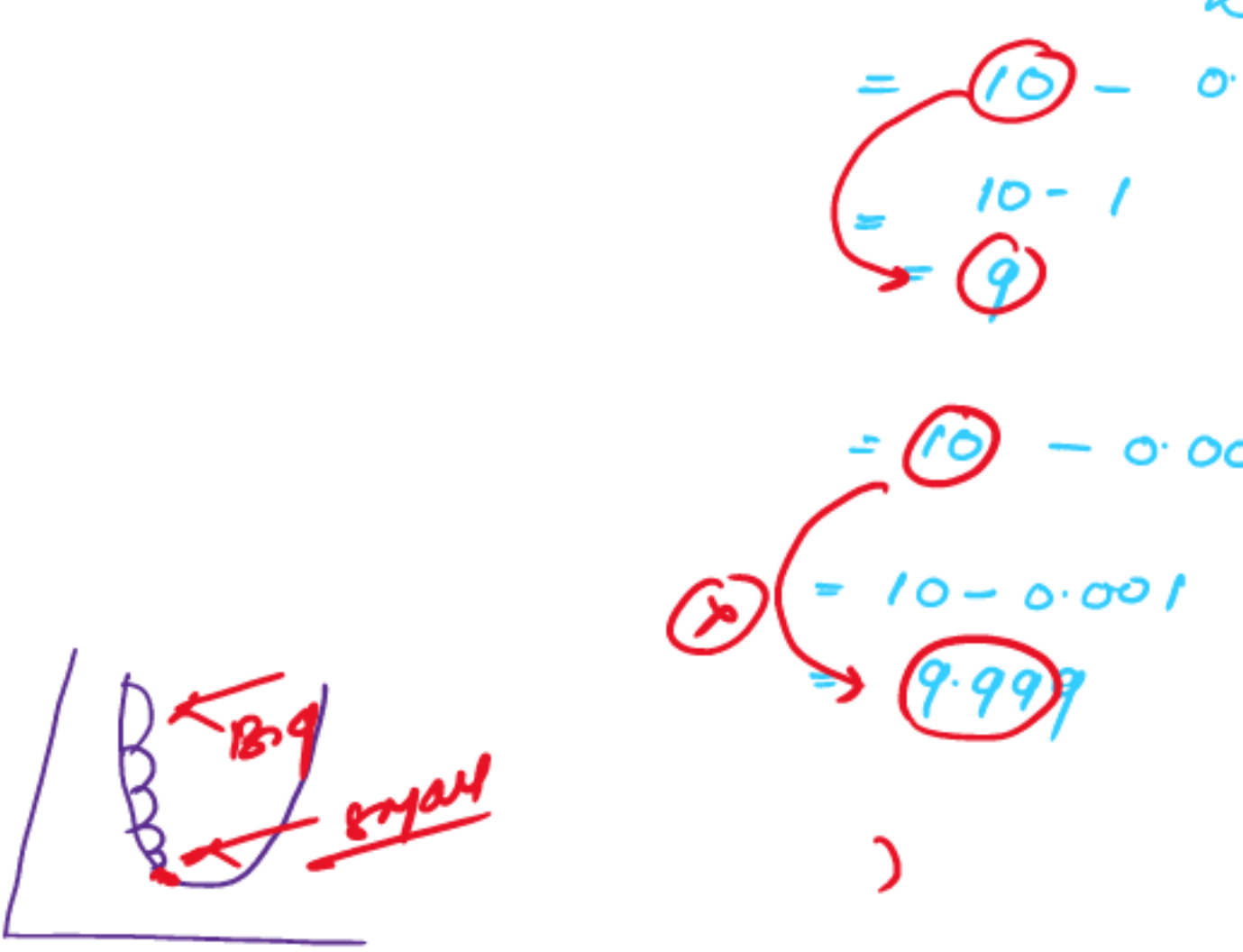
Batch	n_t	α'	
β_1	↑	↑ ↑ ↑ ↑	0.01
β_2	↑ ↑	↑ ↑ ↑	0.008
β_3	↑ ↑ ↑	↑ ↑	0.004
β_4	↑ ↑ ↑ ↑	↑	0.001
	Increasing	Decreasing	

β_1 → 5
β → 10
β_3 → 15 } n_t (↑)

Data → NN → $\hat{y}$ → Loss ← wβα. 0.01
Data → NN → $\hat{y}$ → Loss ← wβα 0.008

β_3

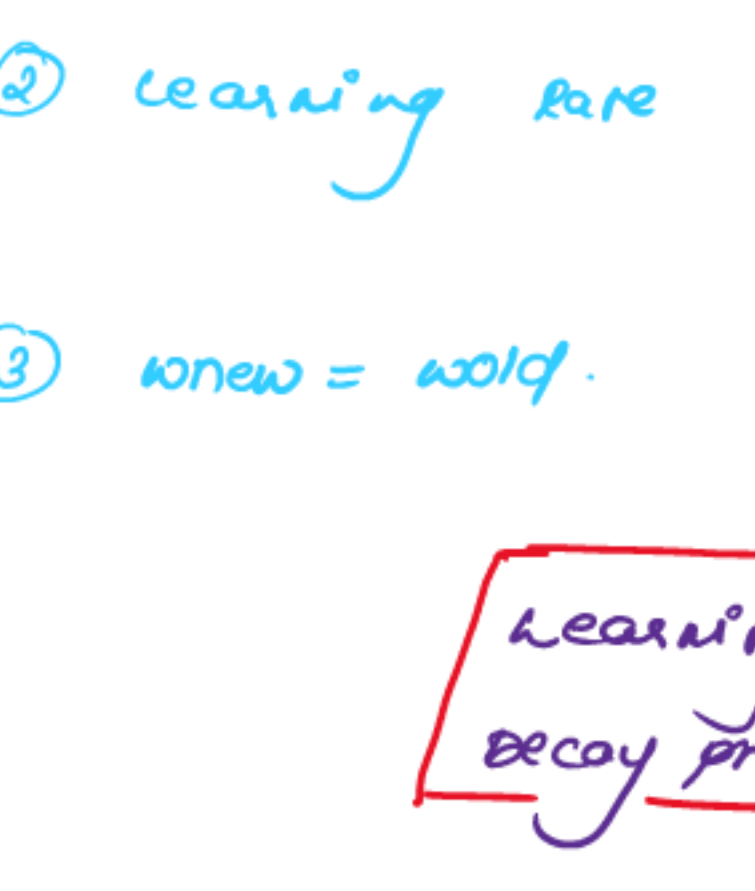
β_4



Fast slow

$= 10 - 0.01 \times 10.$ (α)
$10 - 1$
$= 9$

$= 10 - 0.00001 \times 10$
$= 10 - 0.001$
$= 9.999$



① global minima is not reached in case of Adagrad.

② learning Rate reduces Monotonically (↓)

③ $w_{new} = w_{old}$.

[Learning Rate Decay phenomenon.]
