ICCS482 Deep Learning

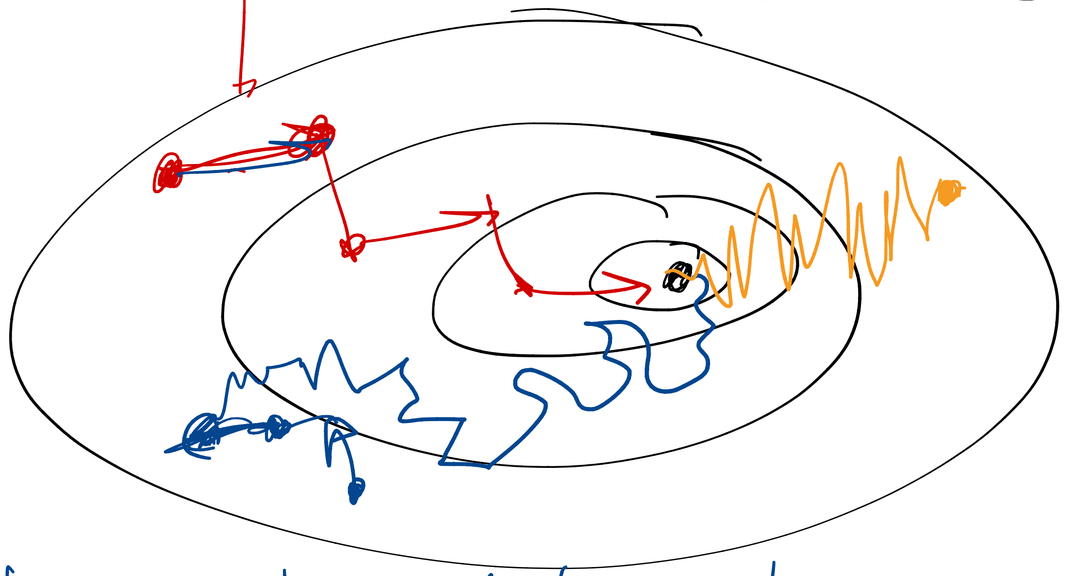
Lecture 6: Optimization

Sunsern Cheamanunkul, Sep 24, 2020.

Gradient Descent

doesn't make use of vectorization

1-step of GD



X is the entire data set

- Divide train data into small batches:

$$\{ X^{\{1\}}, X^{\{2\}}, \ldots X^{\{4\}} \}$$

* update weight even batch

① Stochastic GD

* update weights every example (noisy)
* should not pair with large α

② Batch GD

* update weights after seeing the entire train data.
* If data is not too big, this is ok.

③ Mini-batch GD

batch_size ⟷ # of examples in mini-batch

# of epoch ⟵ # of passes on train data

↳ $\frac{\#\text{ of examples}}{\text{batch\_size}}$ weight updates

batch_size = 1 ⟵ stochastic GD

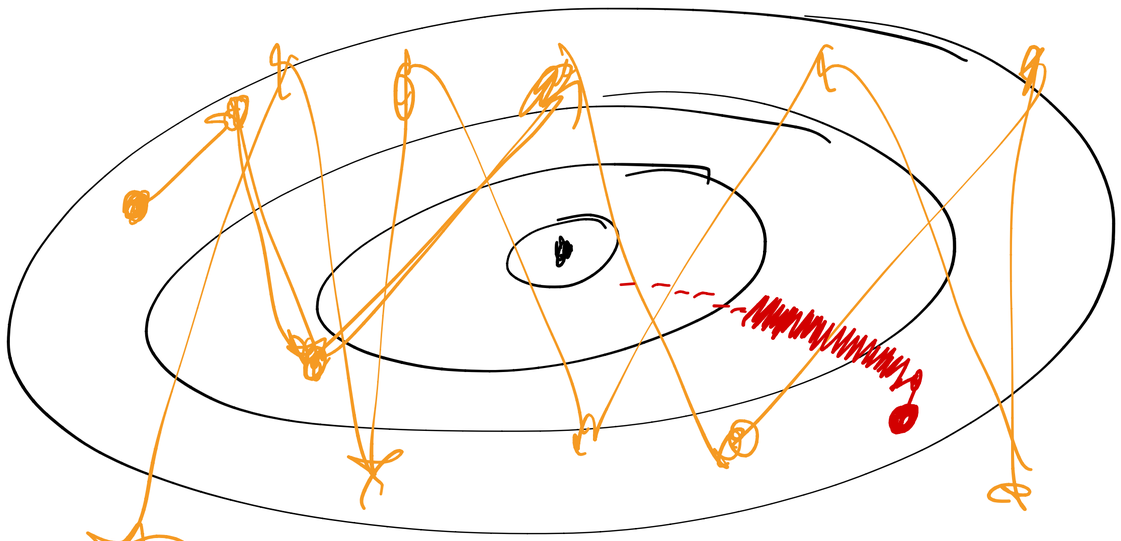
batch_size = m ⟵ batch GD

$2 \leq$ batch_size $\leq m-1$ ⟵ mini-batch GD

In practice,

$$batch\_size = \{32, 64, 128, 256, 512, 1024\}$$

Problem with GD



problem of not converging

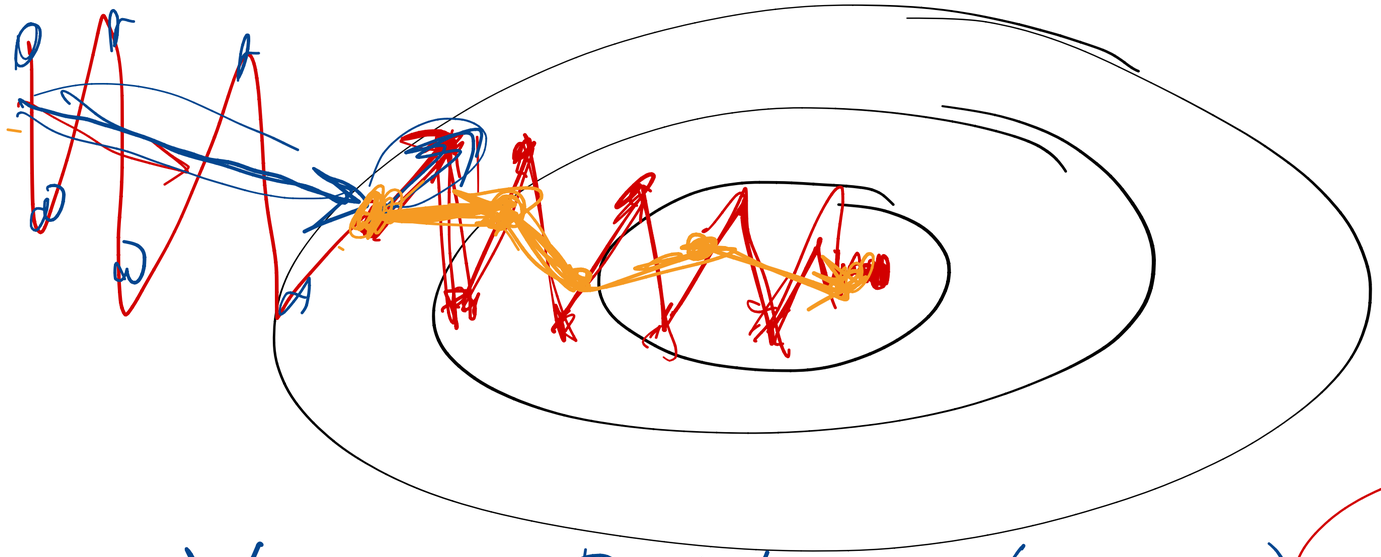
① too high learning rate

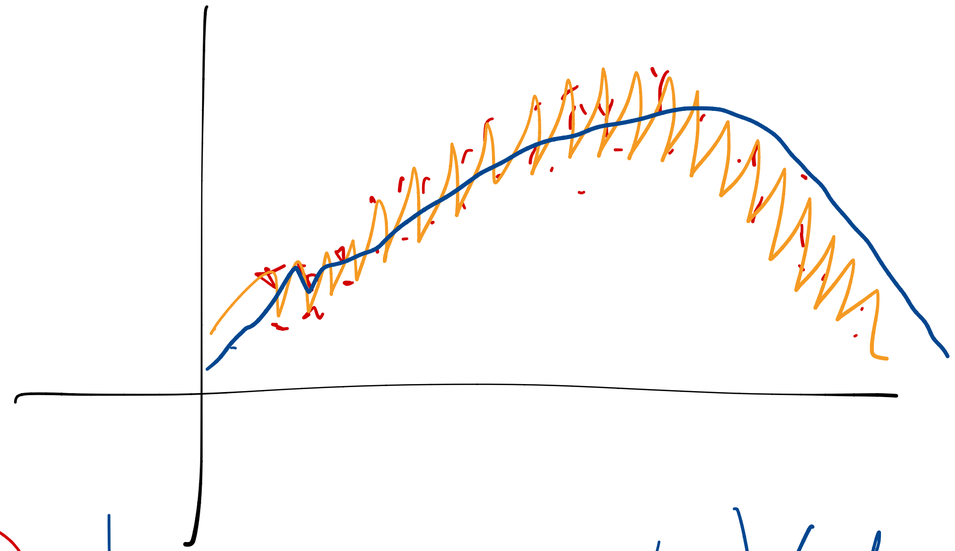
$w = w - \alpha dw$

learning rate

② too low learning rate

Gradient descent with Momentum





$$V_{dw} = \beta V_{dw} + (1-\beta) dw$$

$$V_{db} = \beta V_{db} + (1-\beta) db$$

$$w = w - \alpha \cdot V_{dw}$$

$$b = b - \alpha \cdot V_{db}$$

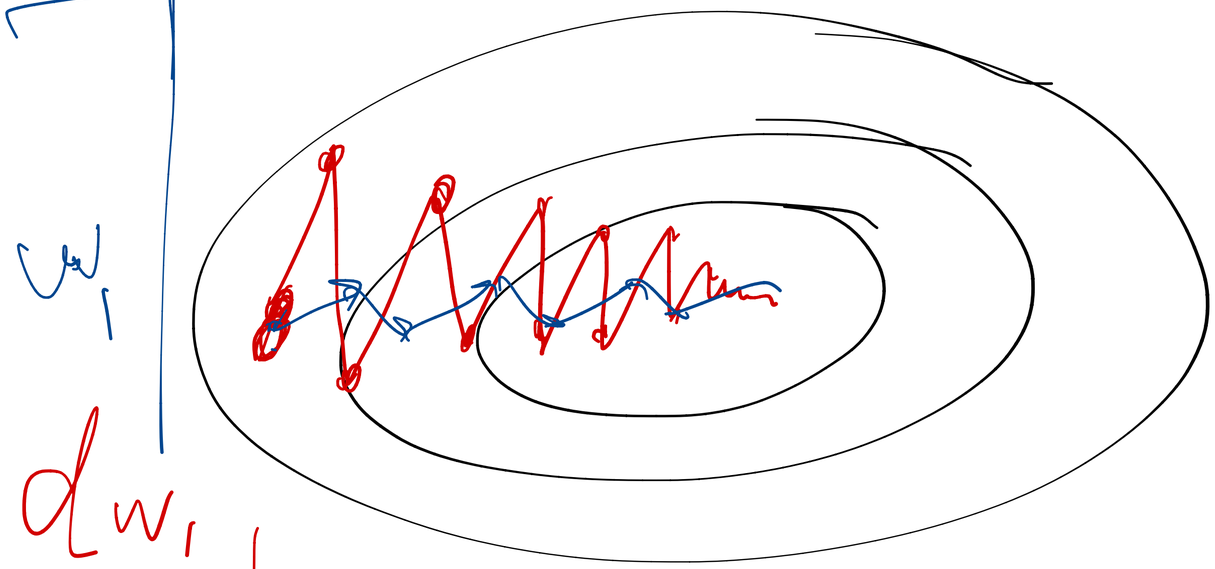
exponential weighted average of dw over iterations.

RMSprop

estimate of variance of each variable

$$S_{dw} = \beta \cdot S_{dw} + (1-\beta) \cdot dw^2$$

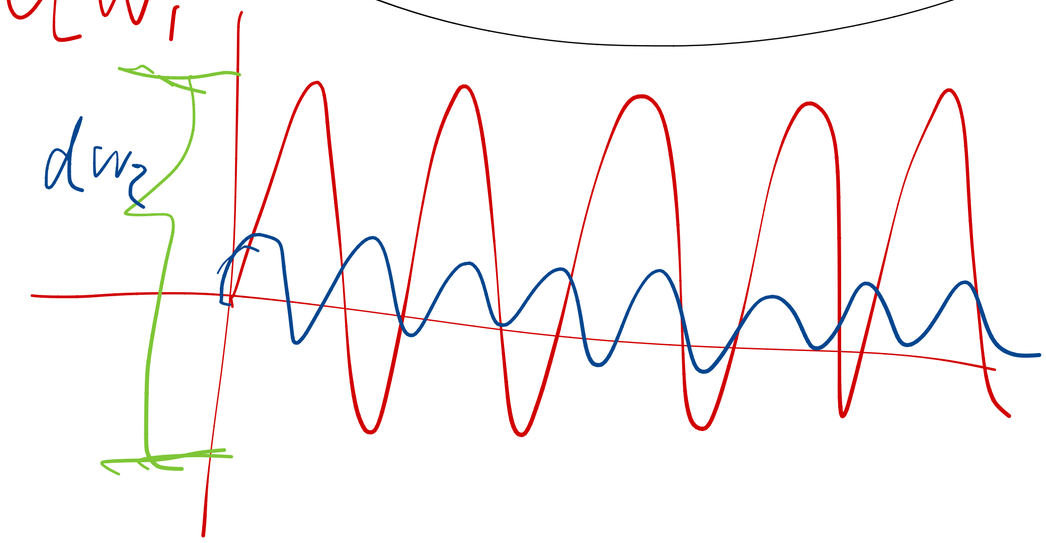
$$S_{db} = \beta \cdot S_{db} + (1-\beta) \cdot db^2$$



$$W = W - \frac{\alpha \cdot dW}{(\sqrt{S_{dw}} + \epsilon)}$$

$$b = b - \frac{\alpha \cdot db}{(\sqrt{S_{db}} + \epsilon)}$$

Auto adjust learning based on correct estimate of variance



$Var(dw_2)$ is smaller than $Var(dw_1)$

iterations

Adam (Adaptive Moment Estimate)

$$V_{dw} = \beta_1 V_{dw} + (1-\beta_1)\, dw$$
$$V_{db} = \beta_1 V_{db} + (1-\beta_1)\, db$$

} 1st moment

$$S_{dw} = \beta_2 S_{dw} + (1-\beta_2)\, dw^2$$
$$S_{db} = \beta_2 S_{db} + (1-\beta_2)\, db^2$$

} 2nd moment

$$w = w - \alpha \cdot \frac{V_{dw}}{\sqrt{S_{dw}} + \epsilon}$$

$$b = b - \alpha \cdot \frac{V_{db}}{\sqrt{S_{db}} + \epsilon}$$

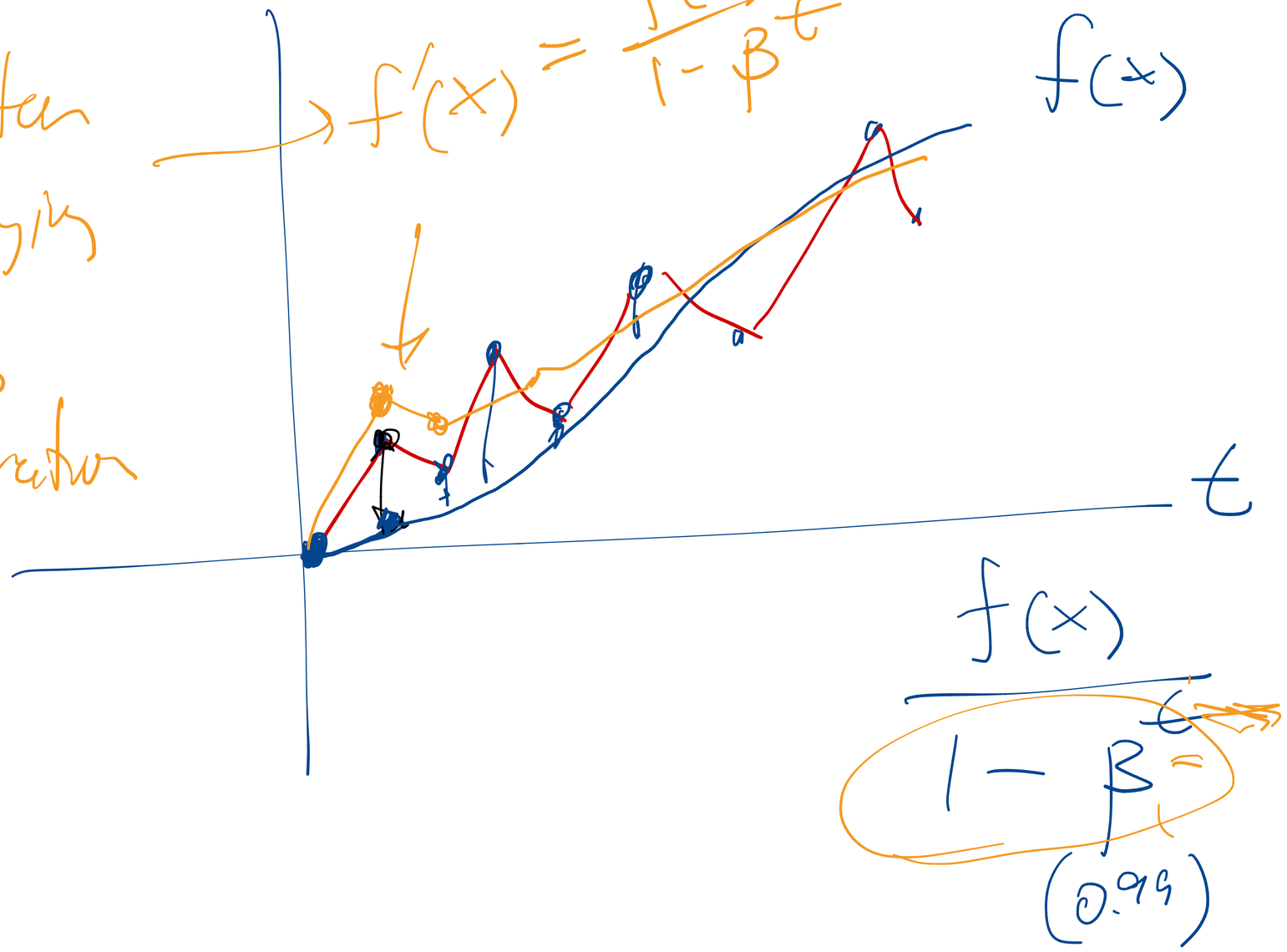
Bias correction term

$$\frac{V_{dw}}{1-\beta_1^t}$$

this corrects cold start problem

after applying bias correction

$f'(x) = \frac{f(x)}{1-\beta^t}$



$\frac{f(x)}{1-\beta^t}$

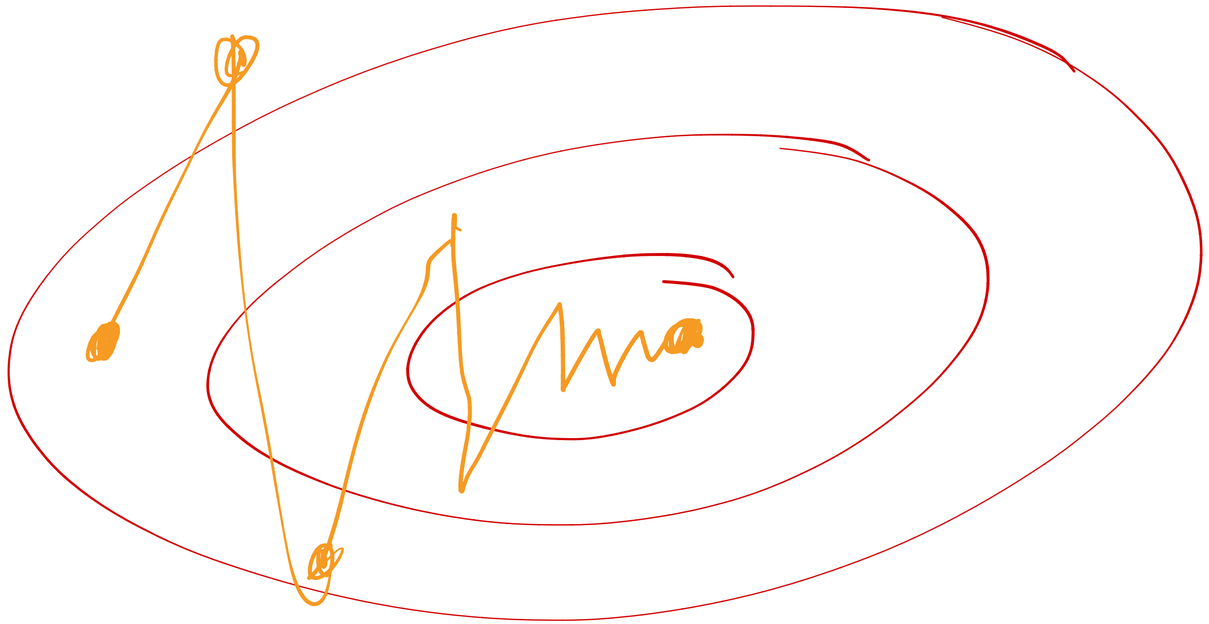
(0.99)

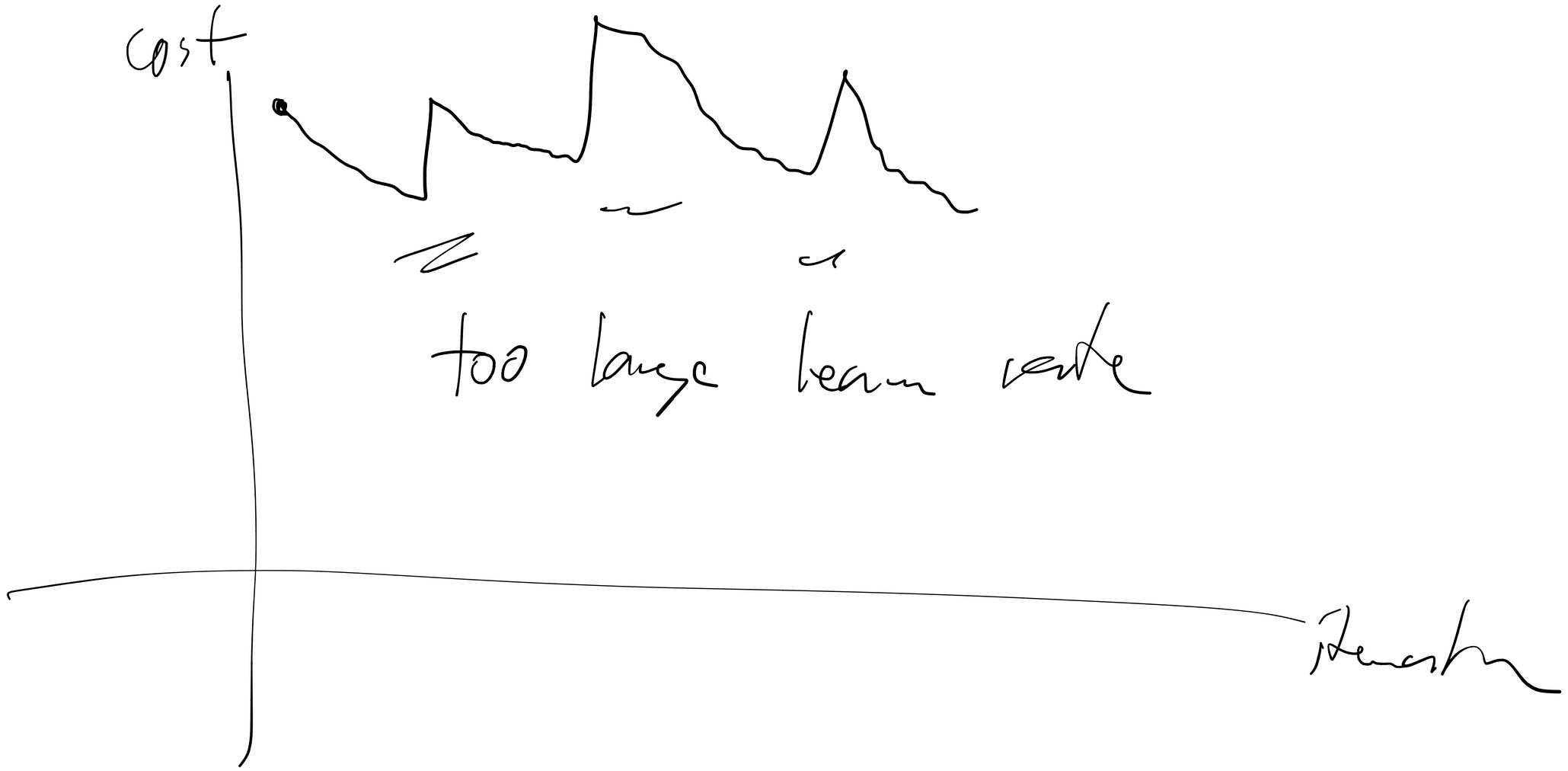
Learning rate decay

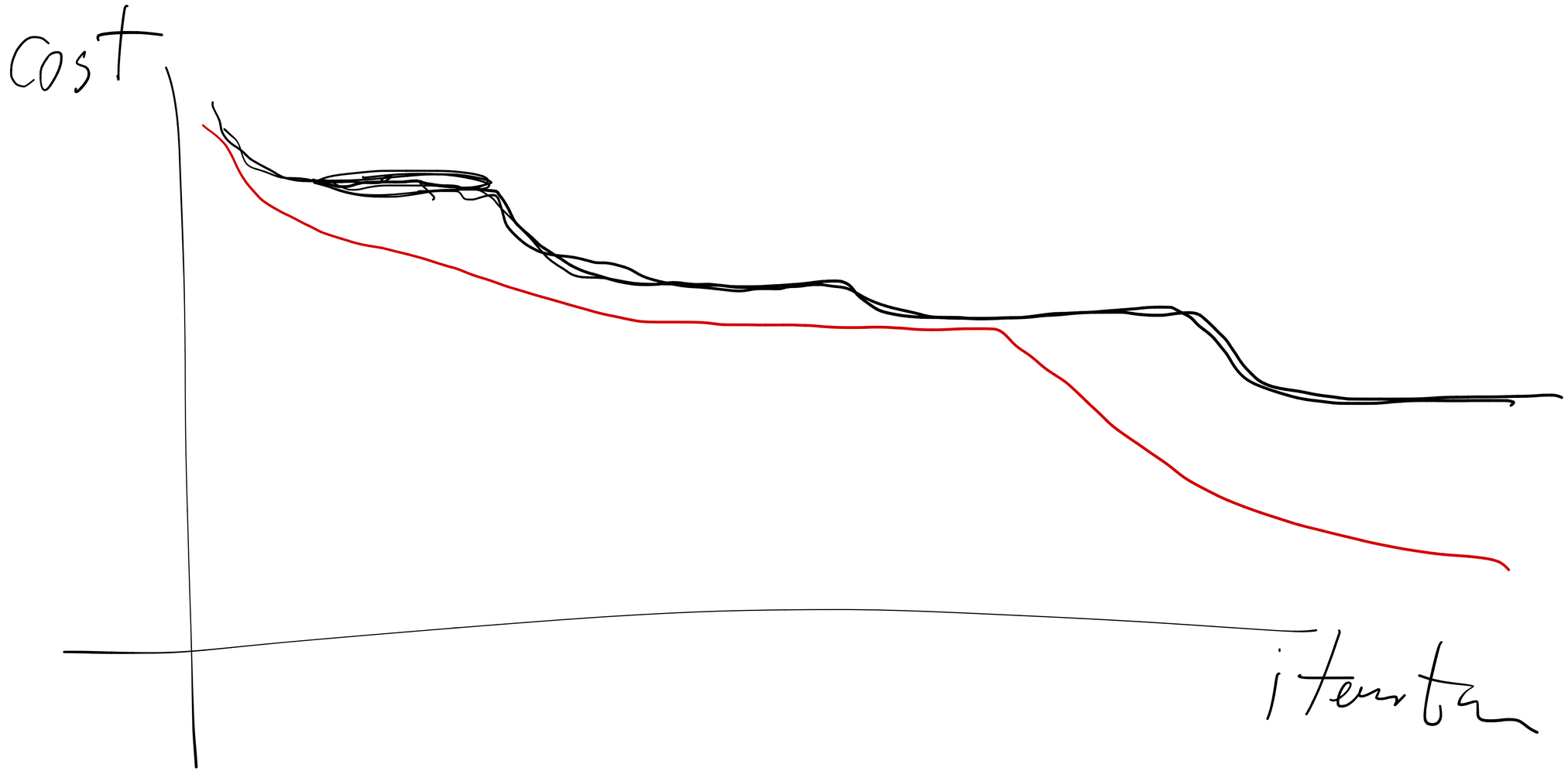
$$\alpha = (0.95^{\text{epoch}}) \cdot \alpha_0$$

initial learning rate

decay-rate




cost
too large learn rate


cost
iteration