



deeplearning.ai

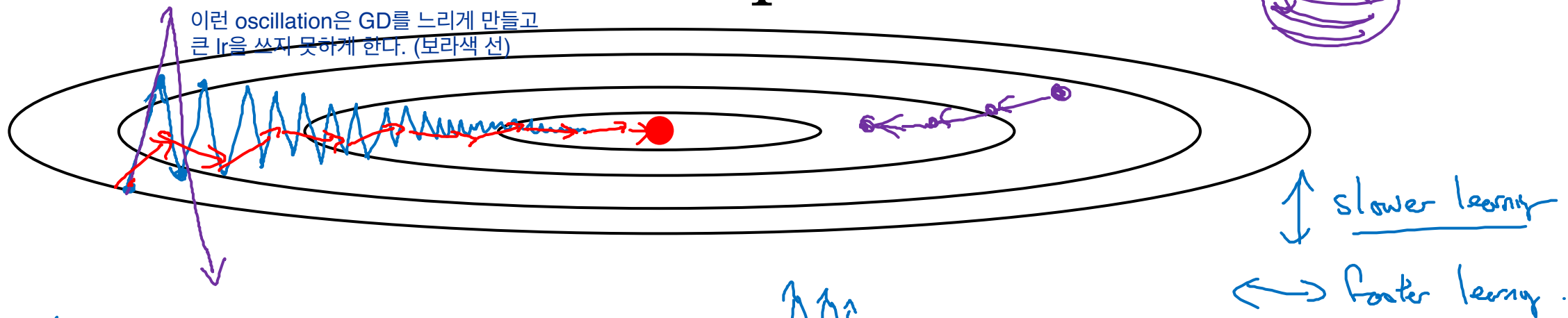
# Optimization Algorithms

---

## Gradient descent with momentum

한마디로, 기본 아이디어는 gradient의 exponentially weighted average를 구해서 weight update 할 때 그 gradient를 사용하는 것이다.

# Gradient descent example



Momentum:

On iteration  $t$ :

Compute  $\Delta W, \Delta b$  on current mini-batch.

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

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

Friction → ↑ velocity

$$W := W - \alpha V_{\Delta W}$$

deep learning ai

$$b := b - \alpha V_{\Delta b}$$

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

# Implementation details

$$v_{dw} = 0, \quad v_{db} = 0$$

On iteration  $t$ :

Compute  $dW, db$  on the current mini-batch

$$\left. \begin{aligned} \rightarrow v_{dW} &= \beta v_{dW} + (1 - \beta) dW \\ \rightarrow v_{db} &= \beta v_{db} + (1 - \beta) db \end{aligned} \right\} \quad \left| \quad \underbrace{v_{dW} = \beta v_{dW} + dW}_{\text{← (1-beta) term을 제거한 이 수식을 사용하는 경우도 많다.}}$$

$$W = W - \underbrace{\alpha v_{dW}}, \quad b = \underline{b} - \underbrace{\alpha v_{db}}$$

$$\cancel{\frac{v_{dW}}{1 - \beta^t}}$$

보통 베타값으로 0.9를 사용하는데 이 경우 지난 10번의 iteration으로 금방 warmup 되므로 대부분의 경우 bias correction은 사용하지 않는다.

Hyperparameters:  $\alpha, \beta$

$$\underline{\beta = 0.9}$$

average over last  $\approx 10$  gradients