Lecture 14. Advanced optimization Methods for DL.

1. Optimization methods in DL. (chapter 3, The science of DL)

① First-order methods

GD/SGD:

$$w^i = w^{i-1} + (-\varepsilon \cdot \nabla_w L)$$

ε: learning rate; $\nabla_w L$: first-order derivative.

② Second-order methods.

Newton's method: $\Delta W_{nst} = -[\nabla^2_{ww} L^{-1}] \cdot \nabla_w L$

newton step:

Quasi-Newtom

$$\nabla^2 L = \begin{bmatrix} \frac{\partial^2 L}{} \end{bmatrix}_{n \times n} \quad L: R^n \to R^n$$

Hessian matrix,

$f: R^n \to R \quad \nabla f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}_{n \times 1}$

$f: R^n \to R^n \quad \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}$

$$\nabla f = \begin{bmatrix} \frac{\partial f^1}{\partial x_1} & \cdots & \frac{\partial f^1}{\partial x_n} \\ \vdots & & \\ \frac{\partial f^n}{\partial x_1} & \cdots & \frac{\partial f^n}{\partial x_n} \end{bmatrix}$$

③ Evolutionary Learning / Genetic algorithms.

fitness function.

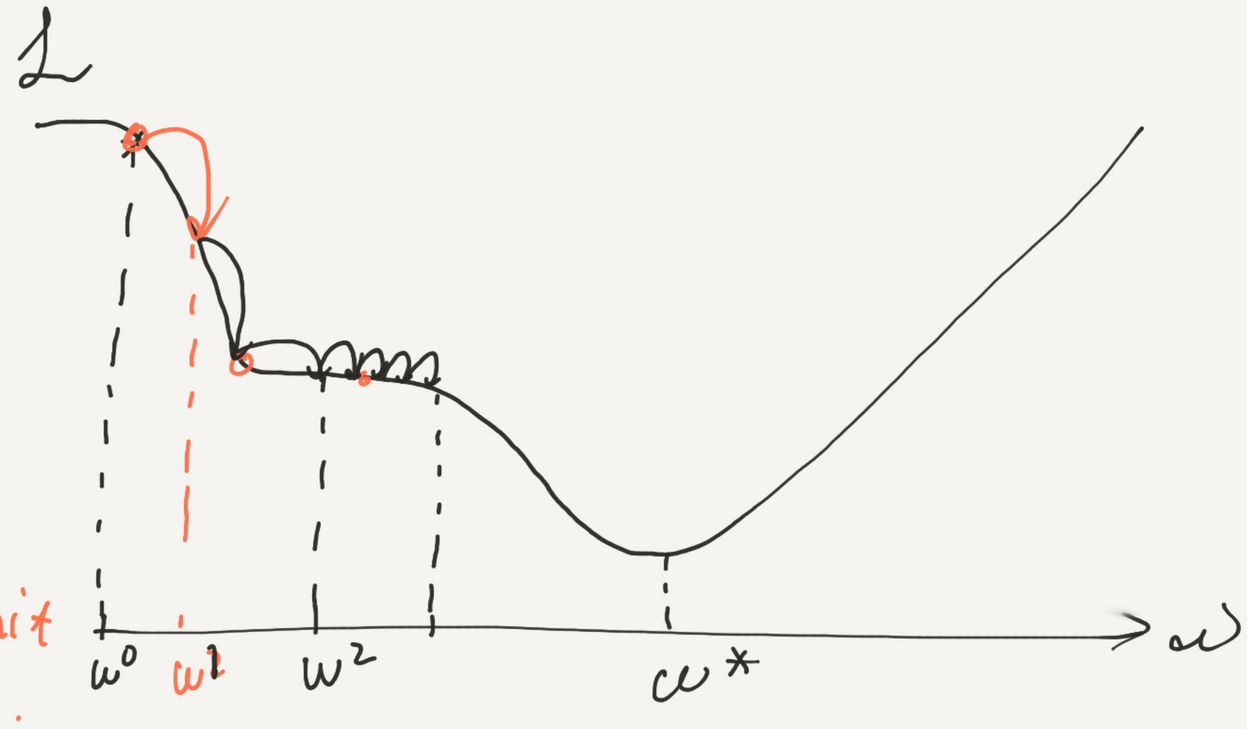
2. Adaptive GD.

GD/SGD:

$w^0, w^1, w^2, \cdots, w^k = w^*$

$w^i = w^{i-1} + (\varepsilon \circ (-\nabla_w L))$

predefined — not unit vector.



step size is determined by both the ε and $\nabla_w L$, if $\nabla_w L$ is very small (flat region), our model converges very slow.

2.1. SGD with momentum.

△ goal is to accelerate learning by accumulating a moving average of past gradients.

SGD: $w^{i+1} = w^i + \underbrace{\varepsilon \cdot (-\nabla_w L)}_{\Delta w_{bp}} = w^i + \Delta w_{bp}$

Δw_{bp} → combined step size

$w^0 \rightarrow \Delta w_{bp}^0 = \varepsilon \cdot (-\nabla L)$

$w^1 \rightarrow \Delta w_{bp}^1$

$w^2 \rightarrow \Delta w_{bp}^2$

$\vdots$

$w^{k-1} \rightarrow \Delta w_{bp}^{k-1}$

moving average : m

$m_0 = \Delta w_{bp}^0$

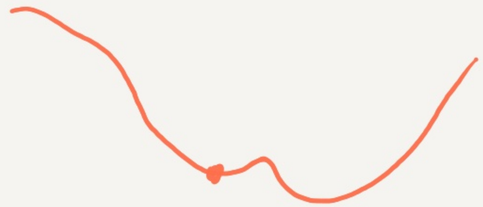
$m_1 = (\Delta w_{bp}^0 + \Delta w_{bp}^1)/2 = \frac{m_0}{2} + \frac{\Delta w_{bp}^1}{2}$

$m_3 = (\Delta w_{bp}^0 + \Delta w_{bp}^1 + \Delta w_{bp}^2)/3$

$= \frac{2 m_1}{3} + \frac{\Delta w_{bp}^2}{3}$

$\vdots$

$m^{k-1} = \frac{(k-2) \cdot m_{k-1}}{k-1} + \frac{\Delta w_{bp}^{k-1}}{k-1}$



SGD with momentum:

$w^{i+1} = w^i + \Delta w^i$

$\Delta w^i = \alpha (\Delta w^{i-1}) + (1-\alpha) \Delta w_{bp}^i$

$0 < \alpha < 1$

2.2 AdaGrad (2011)

It aims to adapts ε by scaling inversely proportional to the square root of accumulated gradients.

Step size: $\Delta w = \frac{1}{\sqrt{r} + \delta} (\varepsilon \circ -1 \cdot \nabla_w L)$

$= \frac{\varepsilon}{\sqrt{r} + \delta} (-\nabla_w L)$

ε → initial learning rate

$\frac{\varepsilon}{\sqrt{r} + \delta}$ → actual learning rate

r : accumulated gradients.

$r^i = r^{i-1} + (\nabla_{w^i} L)^2$ → L^2 norm for vectors.

→ size of the gradient.

$= \sum_{m=1}^{i-1} (\nabla_{w^m} L)^2$

δ : small positive constants $(10^{-7}, 10^{-6})$

to avoid division by zero.

2-3. RMSprop (Hiton, 2012)

Root mean square propagation

RMSprop.

Basic idea: Discard history form extreme past.

$$\Delta W = \frac{\varepsilon}{\sqrt{r+\delta}} \cdot (-\nabla_w L)$$

δ: constant

contribution of g^i's determined by.

$$r^{i+1} = \rho \cdot r^i + (1-\rho) \cdot (\nabla_{w^i} L)^2$$

r^i: historical

ρ: decay rate $\quad 0 < \rho < 1$

$$r^2 = \rho r' + (1-\rho)(\nabla_{w'} L)^2$$

$$r^3 = \rho r^2 + (1-\rho)(\nabla_{w^2} L)^2$$

$$= \rho^2 r' + \rho(1-\rho)(\nabla_{w'} L)^2 + (1-\rho)(\nabla_{w^2} L)^2$$

$$\vdots$$

$$\mu^n = \rho^{n-1} \cdot \mu' + \cdots\cdots$$

$0.9 = \rho$,

$0.81 = \rho^2$

$0.729 = \rho^3$

$0.65 \simeq \rho^4$

$\vdots$

ρ^n

$\rho > \rho^2 > \rho^3$