Deep Learning

Dr. Mehran Safayani
safayani@iut.ac.ir

safayani.iut.ac.ir

https://www.aparat.com/mehran.safayani

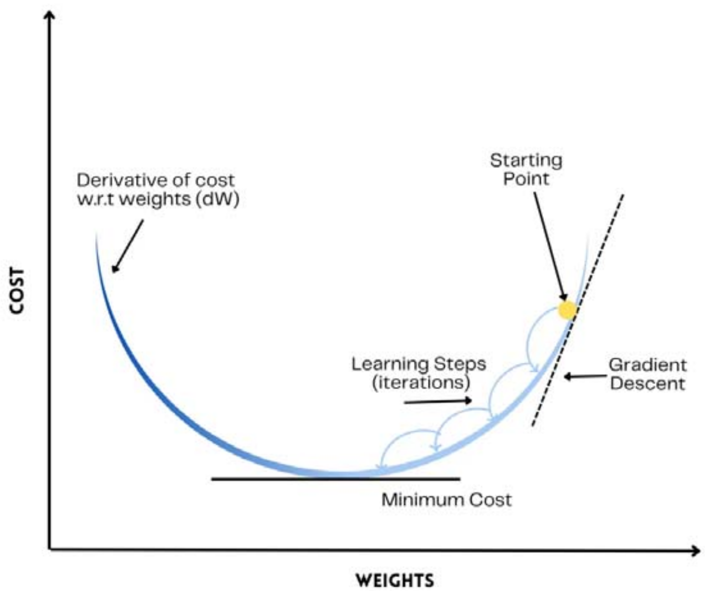
https://github.com/safayani/deep_learning_course

Department of Electrical and computer engineering, Isfahan university of technology, Isfahan, Iran

Basics of Neural Network Programming

Gradient Descent

Gradient Descent



- 1) $\alpha > 0$

Repeat{

$$w = w - \alpha \underbrace{\frac{dJ(w)}{d(w)}}_{dw}$$

}until convergence

$$w = w\text{-}\ \alpha dw$$

$$w^* = (\underbrace{x^T x}_{n^3})^{-1} x^T y$$

$$y = (x - 1)^2$$

$$\frac{dy}{dx} = 2(x - 1) = 0$$

$$x = 1$$

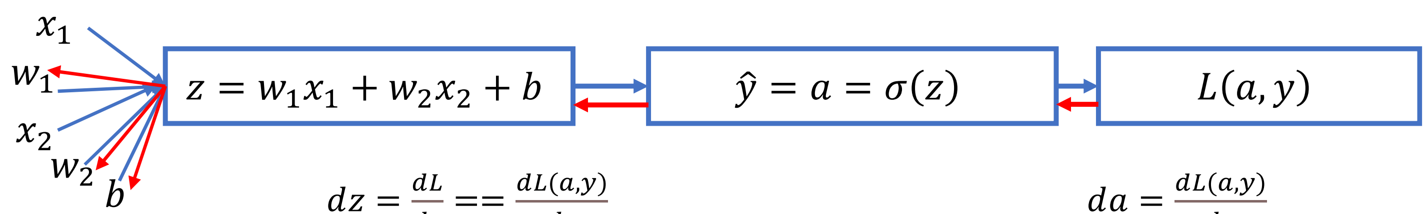
- $z = w^T x + b$
- $\hat{y} = a = \sigma(z)$
- $L(a, y) = -(y \log a + (1 - y) \log(1 - a))$

https://www.askpython.com/

Gradient Descent

Computational Graph

$$da = \frac{dL}{da} = \frac{-y}{a} + \frac{1-y}{1-a} = \frac{a-y}{a(1-a)}$$



$$dz = \frac{dL}{dz} == \frac{dL(a,y)}{dz}$$

$$da = \frac{dL(a,y)}{da}$$

$$dz = a - y_{a(1-a)}$$

$$dz = \frac{dL}{da} \times \frac{\overbrace{da}}{dz} = \frac{dL}{dz}$$

$$a = \sigma(z)$$

$$\frac{da}{dz} = \acute{\sigma}(z) = \underbrace{\sigma(z)}_{a}\,(\underbrace{1 - \sigma(z)}_{1-a})$$

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

$$\frac{\mathrm{d}L}{\mathrm{d}a} = \mathrm{d}a = \frac{a-y}{a(1-a)} \times a(1-a) = a - y$$

$$dw_1 = \frac{dL}{dw_1} = \frac{dL}{\underbrace{dz}_{dz}} \times \frac{dz}{\underbrace{dw_1}_{x_1}} = x_1\,dz$$

$$dw_2 = x_2\,dz$$

$$db = \frac{dL}{db} = \frac{dL}{dz} \times \frac{\overbrace{dz}^{1}}{db} = dz$$

Gradient Descent

- $\begin{cases} w_1 = w_1 - \alpha\, dw_1 \\ w_2 = w_2 - \alpha\, dw_2 \\ b = b - \alpha\, db \end{cases}$ بروزرسانی همزمان

- $\begin{cases} w_1 temp = w_1 - \alpha\, dw_1 \\ w_2 temp = w_2 - \alpha\, dw_2 \\ btemp = b - \alpha\, db \end{cases}$

- $\begin{cases} w_1 = w_1 temp \\ w_2 = w_2 temp \\ b = btemp \end{cases}$

- $J(w,b) = \frac{1}{\mathrm{m}} \sum_{\mathrm{i}=1}^{\mathrm{m}} \mathrm{L}\left(a^{(\mathrm{i})}, \mathrm{y}^{(\mathrm{i})}\right)$

- $a^{(i)} = \hat{y}^{(i)} = \sigma\left(z^{(i)}\right) = \sigma\left(wx^{(i)} + b\right)$

- $dw_1^{(i)} \qquad dw_2^{(i)} \qquad db^{(i)}$

- $\underbrace{\mathrm{d}J(w,b)}_{dw_1} = \frac{1}{\mathrm{m}} \sum_{\mathrm{i}=1}^{\mathrm{m}} \underbrace{\frac{d\mathrm{L}\left(a^{(\mathrm{i})},\mathrm{y}^{(\mathrm{i})}\right)}{dw_1}}_{dw_1^{(i)}}$

Logistic regression on m examples

$J = 0$; $dw_1 = 0$; $dw_2 = 0$; $db = 0$;

$w_1 \leftarrow random$ $w_2 \leftarrow random$ $b \leftarrow random$

Repeat{

For i=1 to m

$z^{(i)} = w^T x^{(i)} + b$

$a^{(i)} = \sigma\left(z^{(i)}\right)$

$J \mathrel{+}= \left[y^{(i)} Log\, a^{(i)} + \left(1 - y^{(i)}\right) Log(1 - a^{(i)})\right]$

$dz^{(i)} = a^{(i)} - y^{(i)}$

$dw_1 \mathrel{+}= x_1^{(i)}\, dz^{(i)}$

$dw_2 \mathrel{+}= x_2^{(i)}\, dz^{(i)}$

$db \mathrel{+}= dz^{(i)}$

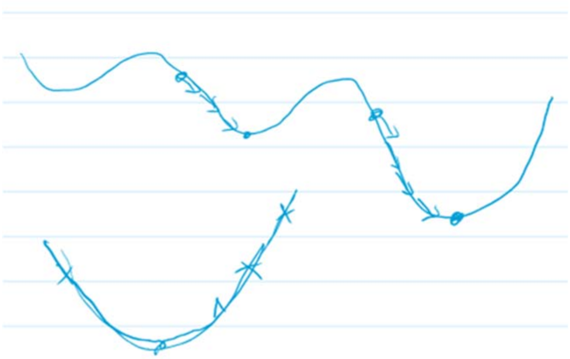
$J /= m$; $dw_1 /= m$; $dw_2 /= m$;

$w_1 = w_1 - \alpha\, dw_1$

$w_2 = w_2 - \alpha\, dw_2$

$b = b - \alpha\, db$

$d\theta = \begin{bmatrix} dw_1 \\ dw_2 \\ db \end{bmatrix}$ $\theta^t = \begin{bmatrix} w_1^t \\ w_2^t \\ b^t \end{bmatrix}$



} until convergence

$\|d\theta\| \le \varepsilon = 10^{-4}$ $\theta^{t+1} = \begin{bmatrix} w_1^{t+1} \\ w_2^{t+1} \\ b^{t+1} \end{bmatrix}$ $\|\theta^{t+1} - \theta^t\|_2 \le \varepsilon$