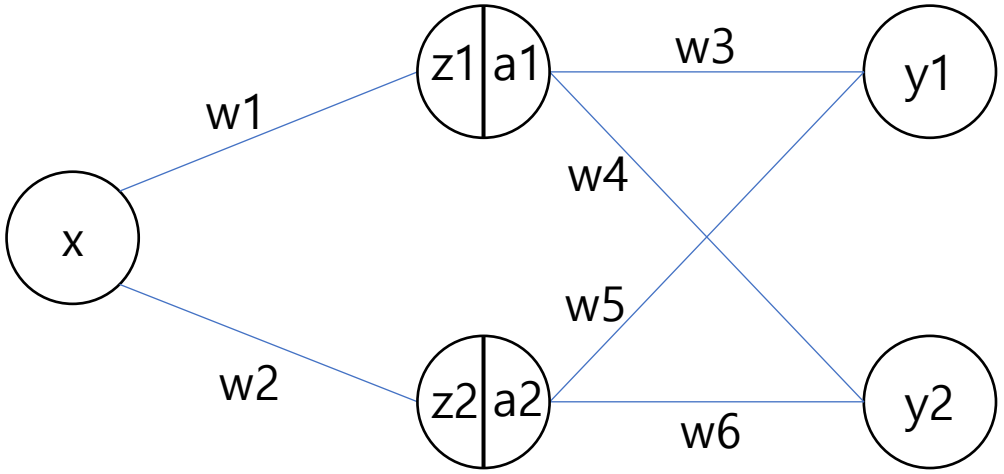
딥알못 탈출하기 #2

고려대학교 지능시스템 연구실
유민형 제작

Gradient descent on matrix

- 지난번 신경망은 총 8개(weight 6개, bias 2개)의 파라미터를 가진다. 8개의 파라미터를 업데이트하기 위해 그래디언트를 구하는 과정에서 체인룰을 이용한 8개의 수식이 필요했다. 기억나는가?



$$\frac{dL}{dw_1} = \frac{dL}{dy_1} \times \frac{dy_1}{da_1} \times \frac{da_1}{dz_1} \times \frac{dz_1}{dw_1} + \frac{dL}{dy_2} \times \frac{dy_2}{da_1} \times \frac{da_1}{dz_1} \times \frac{dz_1}{dw_1}$$

$$\frac{dL}{dw_2} = \frac{dL}{dy_1} \times \frac{dy_1}{da_2} \times \frac{da_2}{dz_2} \times \frac{dz_2}{dw_2} + \frac{dL}{dy_2} \times \frac{dy_2}{da_2} \times \frac{da_2}{dz_2} \times \frac{dz_2}{dw_2}$$

$$\frac{dL}{dw_3} = \frac{dL}{dy_1} \times \frac{dy_1}{dw_3} + \frac{dL}{dy_2} \times \frac{dy_2}{dw_3}$$

$$\frac{dL}{dw_4} = \frac{dL}{dy_1} \times \frac{dy_1}{dw_4} + \frac{dL}{dy_2} \times \frac{dy_2}{dw_4}$$

$$\frac{dL}{dw_5} = \frac{dL}{dy_1} \times \frac{dy_1}{dw_5} + \frac{dL}{dy_2} \times \frac{dy_2}{dw_5}$$

$$\frac{dL}{dw_6} = \frac{dL}{dy_1} \times \frac{dy_1}{dw_6} + \frac{dL}{dy_2} \times \frac{dy_2}{dw_6}$$

$$\frac{dL}{db_1} = \frac{dL}{dy_1} \times \frac{dy_1}{da_1} \times \frac{da_1}{dz_1} \times \frac{dz_1}{db_1} + \frac{dL}{dy_2} \times \frac{dy_2}{da_1} \times \frac{da_1}{dz_1} \times \frac{dz_1}{db_1}$$

$$\frac{dL}{db_2} = \frac{dL}{dy_1} \times \frac{dy_1}{da_2} \times \frac{da_2}{dz_2} \times \frac{dz_2}{db_2} + \frac{dL}{dy_2} \times \frac{dy_2}{da_2} \times \frac{da_2}{dz_2} \times \frac{dz_2}{db_2}$$

12ymh12@gmail.com

Gradient descent on matrix

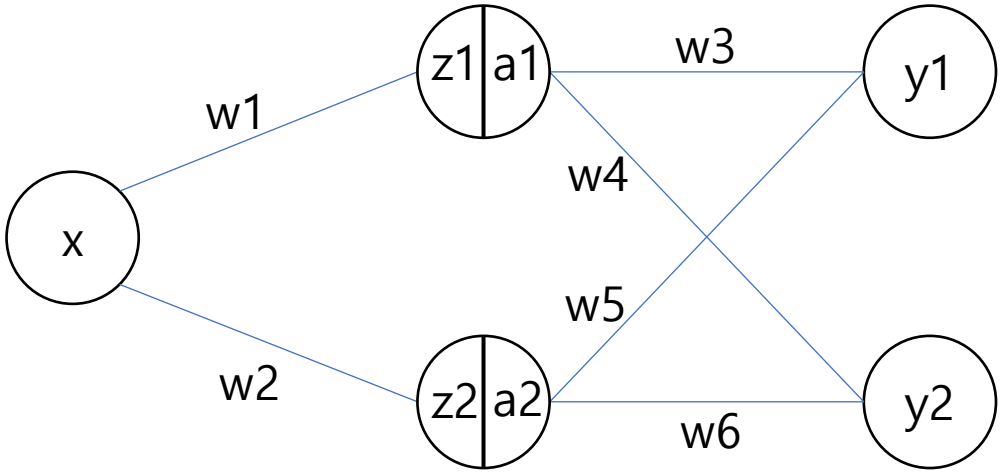
- 근래의 유명한 신경망들은 파라미터의 수가 백 만개를 쉽게 넘어간다. 그렇다면 이 신경망을 구성하고 Gradient Descent를 하기 위해서 컴퓨터는 백 만개의 수식을 일일이 구하는 것일까?

- 이번 회차에서는 수식을 하나하나 구하는 것이 아닌 행렬을 이용하여 forward와 backward를 구성하는 방법을 배우고, 이를 파이썬 numpy 라이브러리로 구현해보자.

12ymh12@gmail.com

Gradient descent on matrix

- (문제1)지난번의 그 신경망을 행렬 단위로 업데이트 하는 방법을 설명하시오. W_1과 W_2, Y, B 행렬로 Forward와 Backward를 표현할 것. 즉, 행렬로 chain rule을 표현하시오.



$$W\_1 = \begin{pmatrix} w1 \\ w2 \end{pmatrix}$$

$$W\_2 = \begin{pmatrix} w3 & w5 \\ w4 & w6 \end{pmatrix}$$

$$Y = \begin{pmatrix} y1 \\ y2 \end{pmatrix}$$

$$B = \begin{pmatrix} b1 \\ b2 \end{pmatrix}$$

$$f(x) = \begin{cases} x \text{ if } x>0 \\ 0 \text{ else} \end{cases}$$

12ymh12@gmail.com

Gradient descent on matrix

- (문제2)위의 신경망을 데이터 (x, y1, y2) = (0.5, 1, -1)로 3번 업데이트 하는 알고리즘을 numpy로 구현하여라

- for i in range(3):

    ## ---- forward ---- ##

    ## ---- backward ---- ##

12ymh12@gmail.com