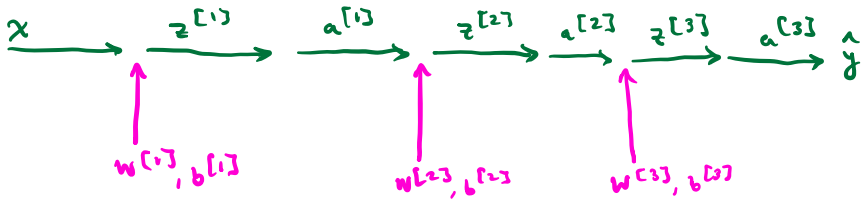
Deep Learning Optimization

Goal: Optimize $w^{[1]}, w^{[2]}, w^{[3]}, b^{[1]}, b^{[2]}, b^{[3]}$



Loss (cost): $J(\hat{y}, y) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}^{(i)}$

where $\mathcal{L}^{(i)} = -\left[y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log (1 - \hat{y}^{(i)}) \right]$

Backpropagation

$\forall l = 1 \dots 3$:

$$w^{[l]} = w^{[l]} - \alpha \frac{\partial J}{\partial w^{[l]}}$$

$$b^{[l]} = b^{[l]} - \alpha \frac{\partial J}{\partial b^{[l]}}$$

$$\frac{\partial J}{\partial w^{[3]}} = \underbrace{\frac{\partial J}{\partial a^{[3]}} \frac{\partial a^{[3]}}{\partial z^{[3]}}}_{\frac{\partial J}{\partial z^{[3]}}} \frac{\partial z^{[3]}}{\partial w^{[3]}}$$

$$\frac{\partial J}{\partial w^{[2]}} = \underbrace{\frac{\partial J}{\partial z^{[3]}} \frac{\partial z^{[3]}}{\partial a^{[2]}} \frac{\partial a^{[2]}}{\partial z^{[2]}}}_{\frac{\partial J}{\partial z^{[2]}}} \frac{\partial z^{[2]}}{\partial w^{[2]}}$$

$$\frac{\partial J}{\partial w^{[1]}} = \frac{\partial J}{\partial z^{[2]}} \frac{\partial z^{[2]}}{\partial a^{[1]}} \frac{\partial a^{[1]}}{\partial z^{[1]}} \frac{\partial z^{[1]}}{\partial w^{[1]}}$$

$$\frac{\partial \mathcal{L}^{(i)}}{\partial w^{[3]}} = -\Big[y^{(i)} \frac{\partial}{\partial w^{[3]}} \log \underbrace{\sigma(w^{[3]} a^{[2]} + b^{[3]})}_{a^{[3]}}$$

$$+ (1-y^{(i)}) \frac{\partial}{\partial w^{[3]}} \log\left(1 - \sigma(w^{[3]} a^{[2]} + b^{[3]})\right)\Big]$$

$$= -\Big[y^{[i]} \frac{1}{a^{[3]}} a^{[3]}(1-a^{[3]}) a^{[2]T}$$

$$+ (1-y^{(i)}) \frac{1}{1-a^{[3]}} (-1)\, a^{[3]}(1-a^{[3]})\, a^{[2]T}\Big]$$

$$= -\left[y^{[i]} - a^{[3]}\right] a^{[2]T}$$

$$\Rightarrow \frac{\partial J}{\partial w^{[3]}} = -\frac{1}{n} \sum_{i=1}^{n} \left[y^{[i]} - a^{[3]}\right] a^{[2]T}$$

$$\frac{\partial \mathcal{L}^{(i)}}{\partial w^{[2]}} = \left(a^{[3]} - y^{(i)}\right) w^{[3]T}\, a^{[2]}(1-a^{[2]})\, a^{[1]T}$$

$$= w^{[3]T}\, a^{[2]}(1-a^{[2]})\left(a^{[3]} - y^{(i)}\right) a^{[1]T}$$

$$\frac{\partial J}{\partial w^{[2]}} = \frac{1}{n} \sum_{i=1}^{n} \frac{\partial \mathcal{L}^{(i)}}{\partial w^{[2]}}$$

Improving Neural Networks

Activation function

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

$$\sigma'(z) = \sigma(z)(1-\sigma(z))$$

$$\text{ReLU}(z) = \begin{cases} 0 & z \le 0 \\ z & z > 0 \end{cases}$$

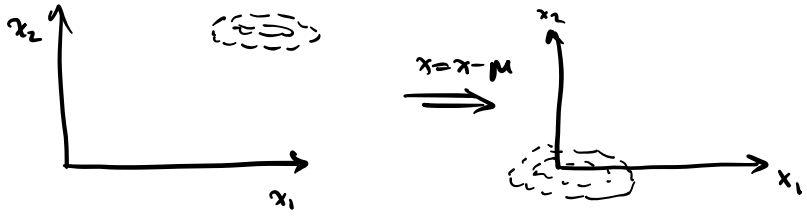
$$\text{ReLU}'(z) = \mathbb{1}_{\{z>0\}}$$

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

$$\tanh'(z) = 1 - \tanh^2(z)$$

Initialization Methods

Normalizing Input



$$\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} = \mu = \frac{1}{n} \sum_{i=1}^{n} x^{(i)}$$

$$\Sigma = \frac{1}{n} \sum_{i=1}^{n} x^{(i)} x^{(i)T}$$

$$x = \Sigma^{-1} x$$

Vanishing / Exploding Gradients

$$\hat{y} = w^{[L]} a^{[L-1]} = w^{[L]} w^{[L-1]} a^{[L-2]}$$

$$\vdots$$

$$= w^{[L]} w^{[L-1]} \cdots w^{[1]} x$$

Gradients can become very large or go to 0!

Avoid by initializing weights close to 1