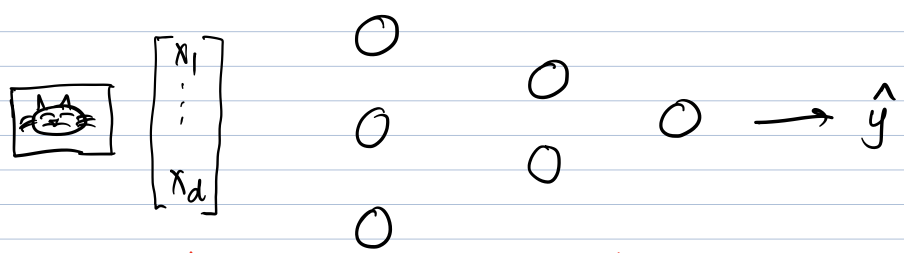
Deep Learning

① Logistic Regression with a NN mindset
② Neural Networks
③ Back propagation
④ Improving your NN



$$\underset{3\times 1}{z^{[1]}} = \underset{3\times d}{w^{[1]}} \underset{d\times 1}{x} + \underset{3\times 1}{b^{[1]}}$$

$$\underset{3\times 1}{a^{[1]}} = \sigma(z^{[1]})$$

$$z^{[2]} = \underset{2\times 3}{w^{[2]}} \underset{3\times 1}{a^{[1]}} + \underset{2\times 1}{b^{[2]}}$$

$$a^{[2]} = \sigma(z^{[2]})$$

$$z^{[3]} = w^{[3]} a^{[2]} + b^{[3]}$$

$$a^{[3]} = \sigma(z^{[3]})$$

$$\hat{y} = a^{[3]}$$

Optimizing $w^{[1]}, w^{[2]}, w^{[3]}, b^{[1]}, b^{[2]}, b^{[3]}$

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

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

$$\frac{\partial J}{\partial w^{[2]}} \qquad \frac{\partial \mathcal{L}^{(i)}}{\partial w^{[2]}}$$

Backward Propagation

$\forall \ell = 1 .. 3$

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

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

$$x \rightarrow z^{[1]} \rightarrow a^{[1]} \rightarrow z^{[2]} \rightarrow a^{[2]} \rightarrow z^{[3]} \rightarrow a^{[3]} \rightarrow \hat{y}$$

$$w^{[1]}, b^{[1]} \qquad w^{[2]}, b^{[2]} \qquad w^{[3]}, b^{[3]}$$

$$\frac{\partial J}{\partial w^{[1]}} \quad \text{vs.} \quad \frac{\partial J}{\partial w^{[3]}}$$

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

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

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

$$\cancel{\frac{\partial J}{\partial w^{[3]}}} \quad \frac{\partial \mathcal{L}^{(i)}}{\partial w^{[3]}} \qquad \mathcal{L}^{(i)} = -\left[y^{(i)} \log \hat{y}^{(i)} + (1-y^{(i)}) \log(1-\hat{y}^{(i)}) \right]$$

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

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

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

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

$$\frac{d}{dx} \log x = \frac{1}{x} \qquad \frac{d}{dx} \sigma(x) = \sigma(x)(1-\sigma(x))$$

$$\frac{\partial \log \sigma(\;)}{\partial w^{[3]}} = \frac{1}{\sigma(\;)} \cdot \frac{\partial \sigma(\;)}{\partial w^{[3]}}$$

$$\frac{\partial \sigma(w^{[3]} a^{[2]} + b^{[3]})}{\partial w^{[3]}} = a^{[3]} (1-a^{[3]}) \cdot \frac{\partial (w^{[3]} a^{[2]} + b^{[3]})}{\partial w^{[3]}}$$

$$\frac{\partial (\overbrace{w^{[3]}}^{1\times 2} \underbrace{a^{[2]}}_{2\times 1} + \overbrace{b^{[3]}}^{1\times 1})}{\partial w^{[3]}} = a^{[2]T}$$

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

$$-y^{(i)} a^{[3]} a^{[2]T} \qquad + y^{(i)} a^{[3]} a^{[2]T}$$

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

$$\frac{\partial J}{\partial w^{[3]}} = -\frac{1}{n} \sum_{i=1}^{n} (y^{(i)} - a^{[3]})\, a^{[2]T}$$

$$\frac{\partial \mathcal{L}}{\partial w^{[2]}} = \frac{\partial \mathcal{L}}{\partial a^{[3]}} \cdot \frac{\partial a^{[3]}}{\partial z^{[3]}} \cdot \frac{\partial z^{[3]}}{\partial a^{[2]}} \cdot \frac{\partial a^{[2]}}{\partial z^{[2]}} \cdot \frac{\partial z^{[2]}}{\partial w^{[2]}}$$

$$\frac{\partial \mathcal{L}}{\partial w^{[3]}} = \overbrace{\frac{\partial \mathcal{L}}{\partial z^{[3]}}}^{a^{[3]} - y^{(i)}} \cdot \overbrace{\frac{\partial z^{[3]}}{\partial w^{[3]}}}^{a^{[2]T}}$$

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

$$= \underbrace{w^{[3]T}}_{(2,1)} * \underbrace{a^{[2]}(1 - a^{[2]})}_{(2,1)} \underbrace{(a^{[3]} - y^{(i)})}_{(1,1)} \cdot \underbrace{a^{[1]T}}_{(1,3)}$$

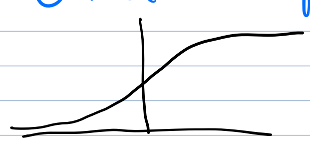
element wise product

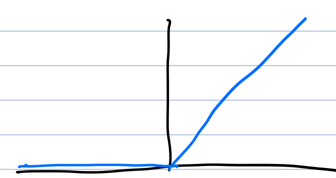
$$\frac{\partial J}{\partial w^{[2]}} = \frac{1}{n} \sum_{i=1}^{n} \frac{\partial \mathcal{L}^{(i)}}{\partial w^{[2]}}$$

Try: compute $\frac{\partial J}{\partial w^{[1]}}$ $\quad \frac{\partial J}{\partial b^{[2]}}$

Improving your NN

(A) Activation functions





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

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

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

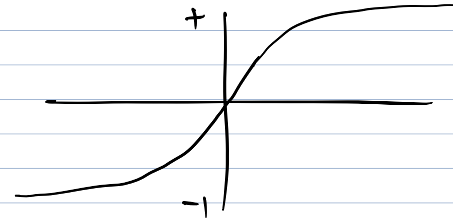
$$\mathrm{ReLU}'(z) = \mathbb{1}\{z > 0\}$$

replaces $a^{[3]}(1 - a^{[3]})$

$a^{[2]}(1 - a^{[2]})$

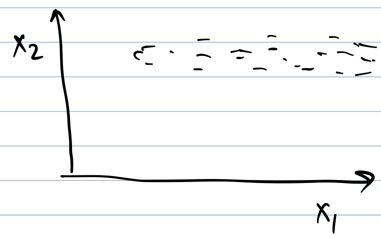
$$\tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

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

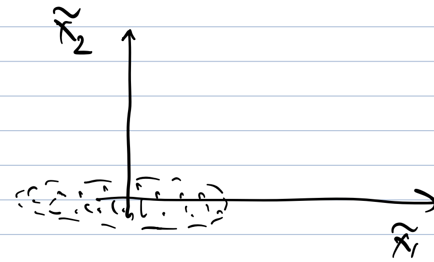


Ⓑ Initialization Methods

Normalizing input



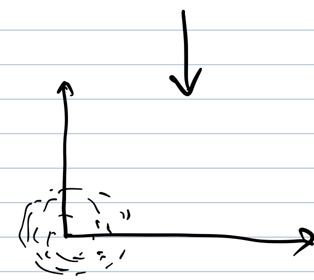
$x = x - \mu$



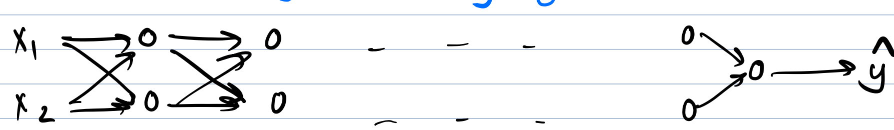
$$\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} = \mu = \frac{1}{n} \sum_{i=1}^{n} x^{(i)}$$

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

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



Vanishing / Exploding gradients



Assume $b = 0$
activation f^n id : $z \to z$

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

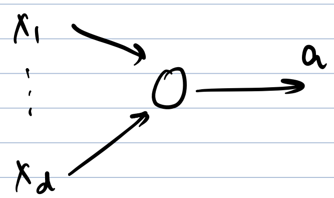
$$= W^{[L]} W^{[L-1]} \cdots W^{[1]} x$$

$$w^{(t)} = \begin{bmatrix} 1.5 & 0 \\ 0 & 1.5 \end{bmatrix} \sim \begin{bmatrix} 1.5^L & 0 \\ 0 & 1.5^L \end{bmatrix}$$

$$\begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \sim \begin{bmatrix} 0.5^L & 0 \\ 0 & 0.5^L \end{bmatrix}$$

— avoid by initializing wts. close to 1

Example w. one neuron



$a = \sigma(z)$

$z = w_1 x_1 + \ldots + w_d x_d$

large d $\longrightarrow$ small w_i

$w_i \sim \frac{1}{d}$

w_L = np.random.randn(shape) * np.sqrt$\left(\frac{1}{n^{[L-1]}}\right)$

for sigmoid

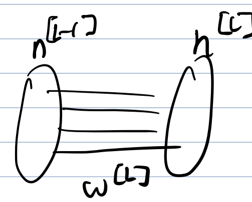
for ReLU : 2 instead of 1

Xavier Initialization

$w^{[L]} \sim \sqrt{\frac{1}{n^{[L-1]}}}$ for tanh

He Initialization

$w^{[L]} \sim \sqrt{\frac{2}{n^{[L]} + n^{[L-1]}}}$

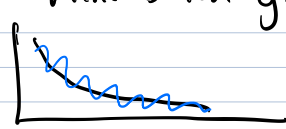


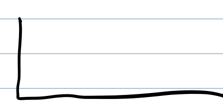
backward prop.   forward prop

Optimization:

Gradient Descent

Mini batch gradient Descent





GD + Momentum

Momentum

$$W = W - \alpha \frac{\partial L}{\partial W}$$

$$W = W - \alpha v$$

$$v = \beta \cdot v + (1-\beta)\frac{\partial L}{\partial W}$$

v : momentum — weighted average of past updates