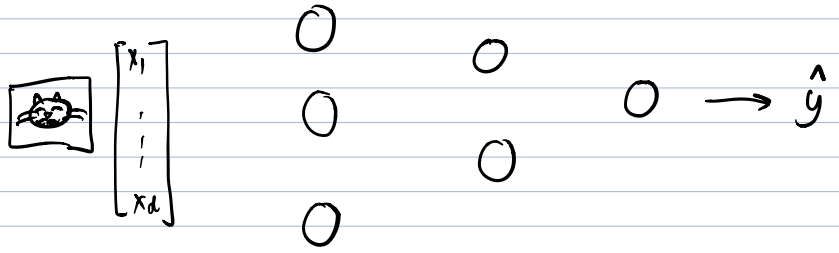
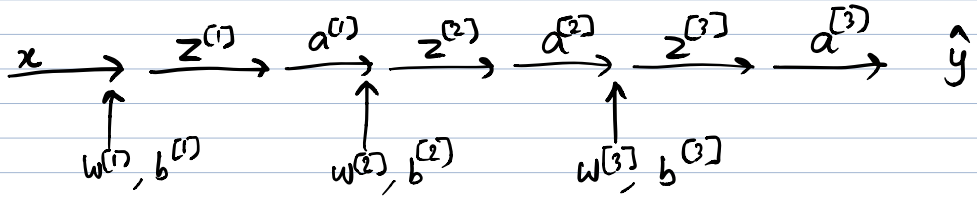
Deep Learning

1. Logistic Regression with a NN mindset
2. Neural Networks
3. Backpropagation
4. Improving your NN



$$z^{[1]} = w^{[1]} x + b^{[1]}$$
$$a^{[1]} = \sigma(z^{[1]})$$
$$z^{[2]} = w^{[2]} a^{[1]} + b^{[2]}$$
$$a^{[2]} = \sigma(z^{[2]})$$
$$z^{[3]} = w^{[3]} a^{[2]} + b^{[3]}$$
$$a^{[3]} = \sigma(z^{[3]})$$
$$\hat{y} = a^{[3]}$$

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



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

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

$\frac{\partial J}{\partial w^{[2]}}$ $\frac{\partial \mathcal{L}}{\partial w^{[2]}}$

Backward Propagation

$\forall l = 1 \ldots 3$

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

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

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

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

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

~~$\frac{\partial J}{\partial w^{[3]}}$~~ $\frac{\partial \mathcal{L}^{(i)}}{\partial w^{[3]}} =$

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

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

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

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

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

$\log' x = \frac{1}{x}$, $\sigma'(x) = \sigma(x)(1-\sigma(x))$

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

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

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

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

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

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

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

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

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

(2,3)

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

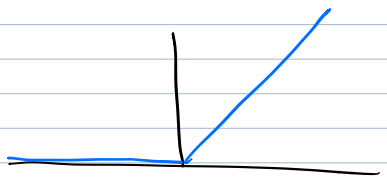
(2,1) $\quad$ (2,1) $\quad$ (1,1) $\quad$ (1,3)

$*$ ↑ element wise product

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

③ Improving your NN

(A) Activation functions



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

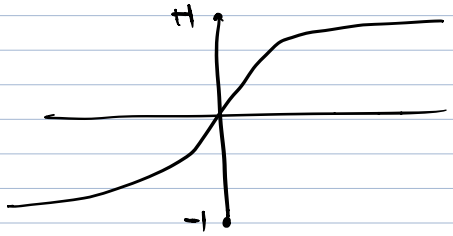
replaces $a^{[3]}(1-a^{[3]})$, $a^{[2]}(1-a^{[2]})$

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

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

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

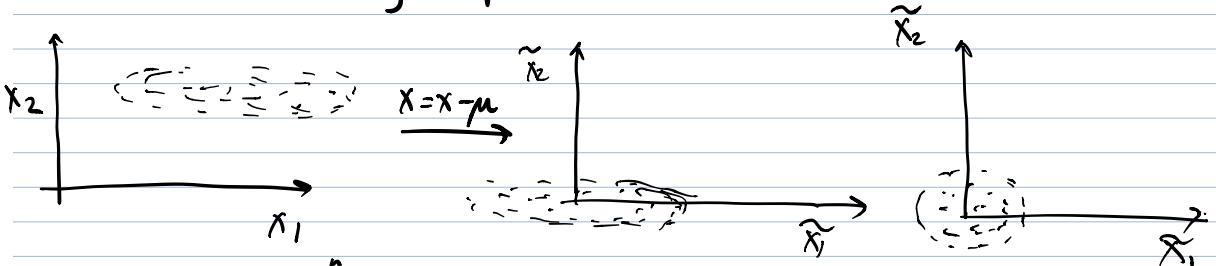
$\tanh(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$



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

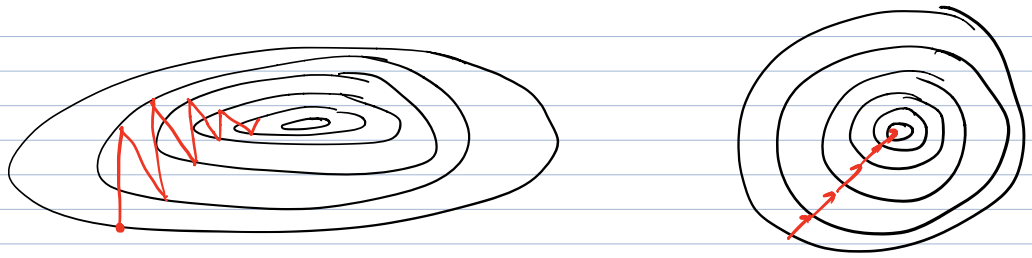
(B) Initialization methods

Normalizing input

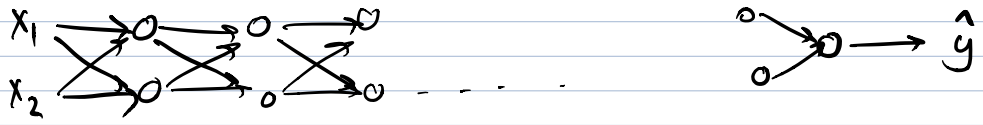


$\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} = \mu = \frac{1}{n} \sum_{i=1}^{n} x^{(i)}$ $\quad \sigma^2 = \frac{1}{n} \sum_{i=1}^{n} (x^{(i)})^2$

$x = \frac{x}{\sigma}$ $\quad x = \Sigma^{-1} x$



Vanishing / Exploding gradients



Assume initialization $b = 0$

Activation fn id $z \mapsto z$

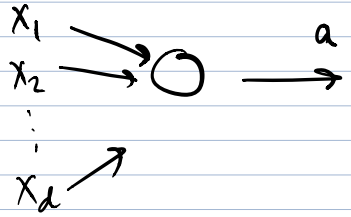
$$\hat{y} = w^{[L]} a^{[L-1]} = w^{[L]} \cdot w^{[L-1]} \cdot a^{[L-2]}$$
$$= w^{[L]} w^{[L-1]} \cdots w^{[1]} \cdot x$$

$$w^{[l]} = \begin{bmatrix} 1.5 & 0 \\ 0 & 1.5 \end{bmatrix} \sim \begin{bmatrix} 1.5^L & 0 \\ 0 & 1.5^L \end{bmatrix}$$

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

— avoid by initializing wts close to 1

Example with 1 neuron



$a = \sigma(z)$

$z = w_1 x_1 + \cdots w_d x_d$

large $d \rightarrow$ small w_i

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

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

for sigmoid

for ReLU : 2 instead of 1

Xavier Initialization

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

He Initialization

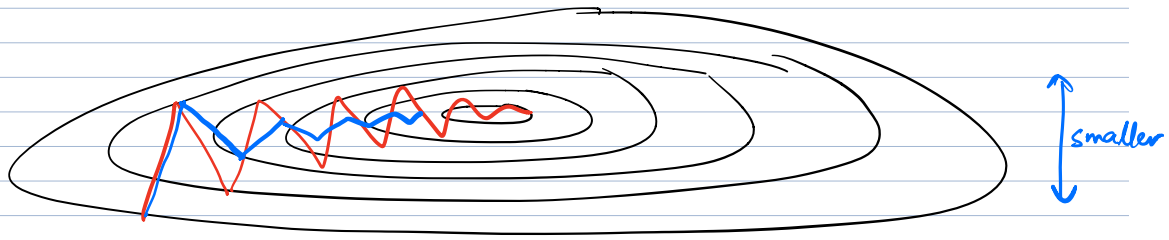
$$w^{[L]} \sim \sqrt{\frac{2}{n^{[L]} + n^{[L-1]}}}$$

$n^{[L]}$: backward prop. $n^{[L-1]}$: forward prop

Gradient Descent

Stochastic Gradient Descent

Mini Batch gradient Descent



Momentum

$$w = w - \alpha \frac{\partial \mathcal{L}}{\partial w}$$

$$v = \beta v + (1-\beta)\frac{\partial \mathcal{L}}{\partial w}$$

$$w = w - \alpha v$$

current gradient : $(1-\beta)$

last gradient : $\beta(1-\beta)$

2nd last : $\beta^2(1-\beta)$

⋮