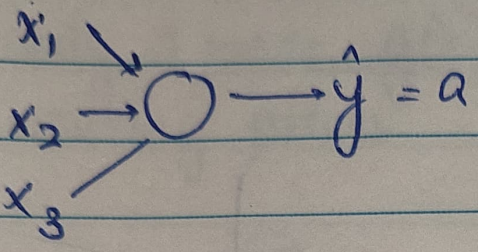
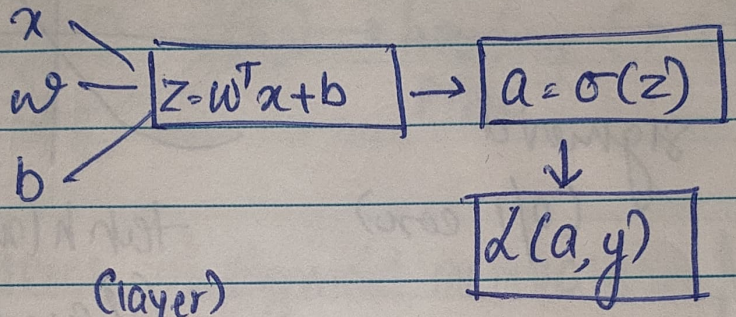
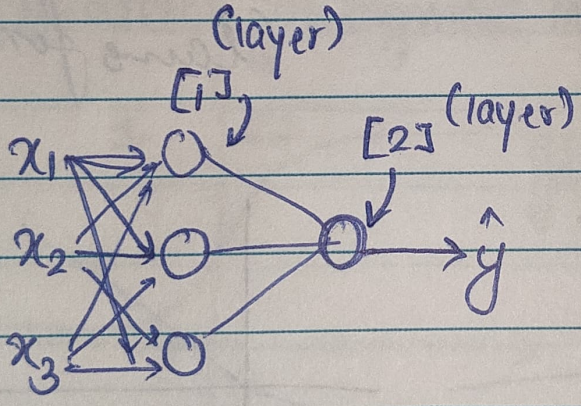
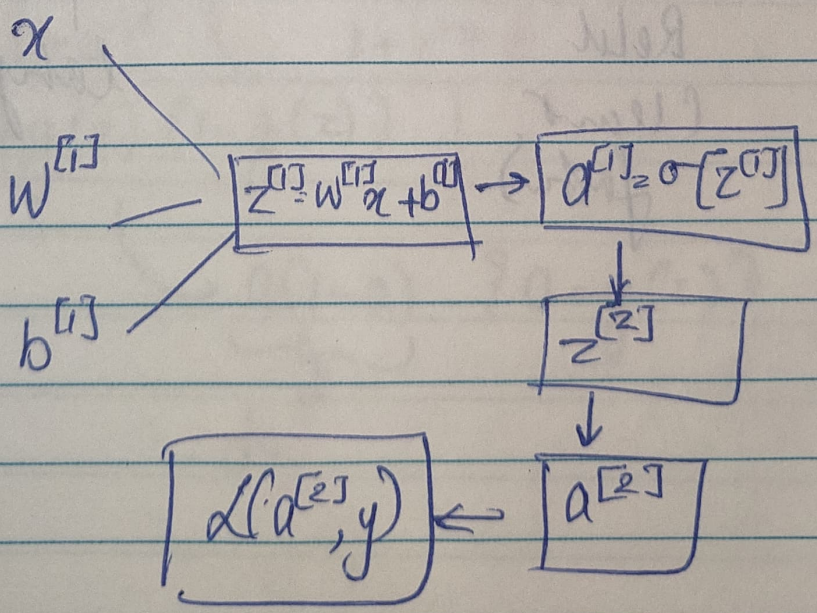
Week 3

Lec-1 Neural Network



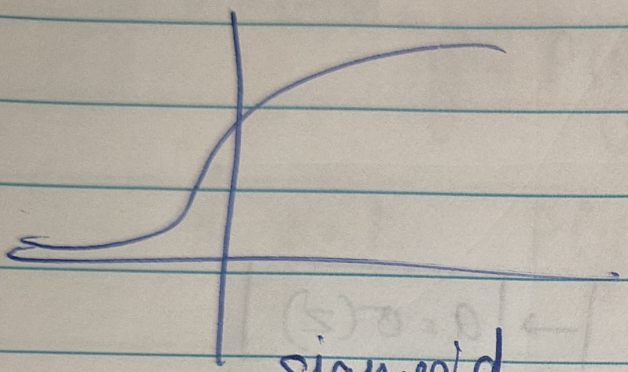




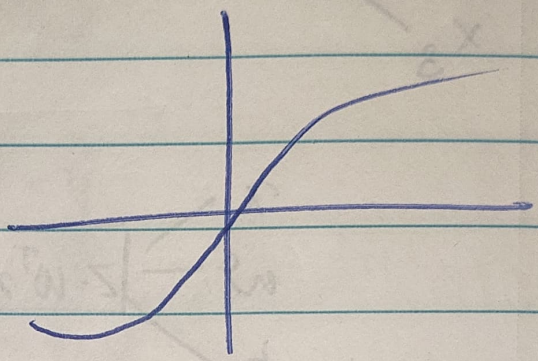


L-2 representation

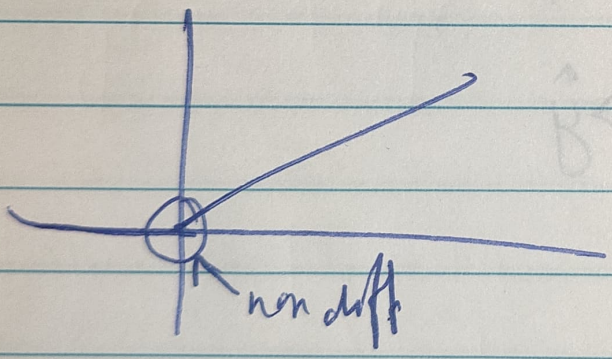
Activation fn.



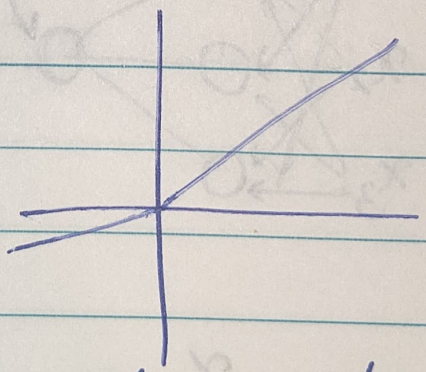
sigmoid
(0/1 conv)



tanh(x)
(learns faster)



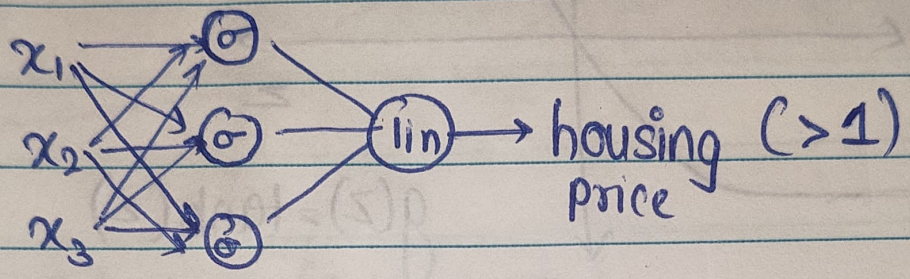
ReLU
(learns faster)



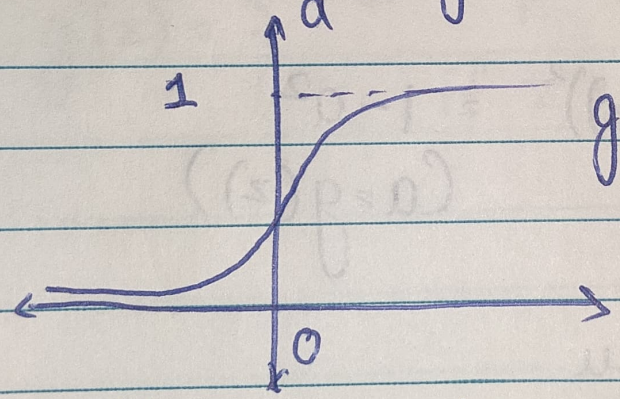
Leaky ReLU

Why non-linear activation functions?

only 1 place $g(z) = z$ used
housing prices



Derivatives of sigmoid activation fns.



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

$$\frac{dg(z)}{dz} = \frac{1}{1+e^{-z}}\left(1 - \frac{1}{1+e^{-z}}\right)$$

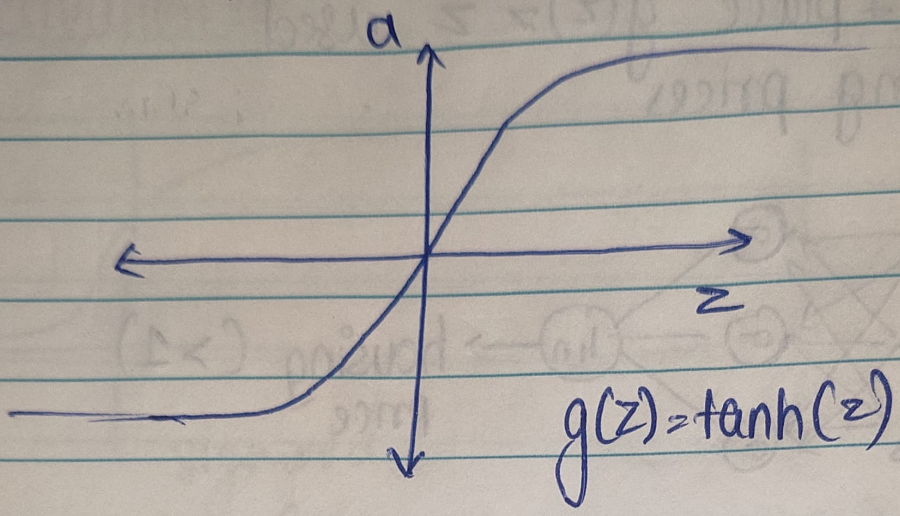
$$= \boxed{g(z)\,(1-g(z))}$$

$g'(z)$

$\rightarrow a(1-a) \quad \{a = g(z)\}$

da

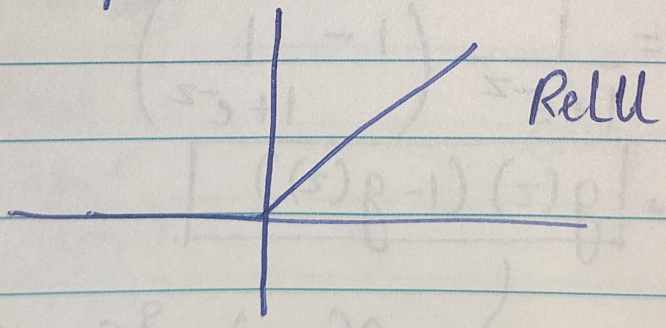
derivative for tanh(x) function



$$\frac{d\,g(z)}{dz} = 1 - (\tan(z))^2$$

$$g'(z) = 1 - (g(z))^2 = 1 - a^2$$
$$(a = g(z))$$

derivative for ReLu

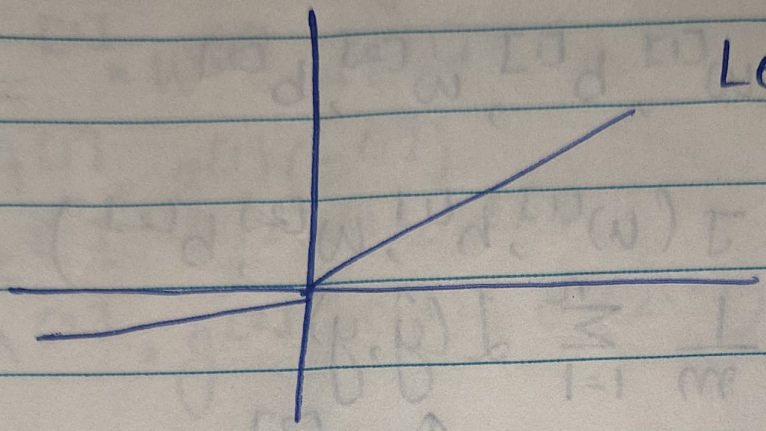


$$g(z) = \max(0, z)$$

$$g'(z) = \begin{cases} 0 & \text{if } z < 0 \\ 1 & \text{if } z > 0 \end{cases}$$

undef. if $z = 0$

derivative for Leaky ReLU



$$g(z) = \max(0.01z, z)$$

$$g'(z) = \begin{cases} 0.01 & \text{if } z < 0 \\ 1 & \text{if } z > 0 \end{cases}$$

gradient descent for neural networks

parameters: $w^{[1]}, b^{[1]}, w^{[2]}, b^{[2]}$

Cost function: $J(w^{[1]}, b^{[1]}, w^{[2]}, b^{[2]})$

$$= \frac{1}{m} \sum_{i=1}^{n} L(\hat{y}, y)$$

$\hat{y} = a^{[2]}$

gradient descent:

Repeat {

Compute predicts $(\hat{y}^{[i]}, i=1, \ldots, m)$

$$dw^{[1]} = \frac{dJ}{dw^{[1]}}, \quad db^{[1]} = \frac{dJ}{db^{[1]}}$$

$$w^{[1]} = w^{[1]} - \alpha\, dw^{[1]}$$

$$b^{[1]} = b^{[1]} - \alpha\, db^{[1]}$$

⋮

}

Forward propagation

$$z^{[1]} = W^{[1]} x + b^{[1]}$$

$$A^{[1]} = g^{[1]}(z^{[1]})$$

$$z^{[2]} = W^{[2]} A^{[1]} + b^{[2]}$$

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

Back propagation

$$dz^{[2]} = A^{[2]} - Y$$

$$dW^{[2]} = \frac{1}{m} dz^{[2]} A^{[1]T}$$

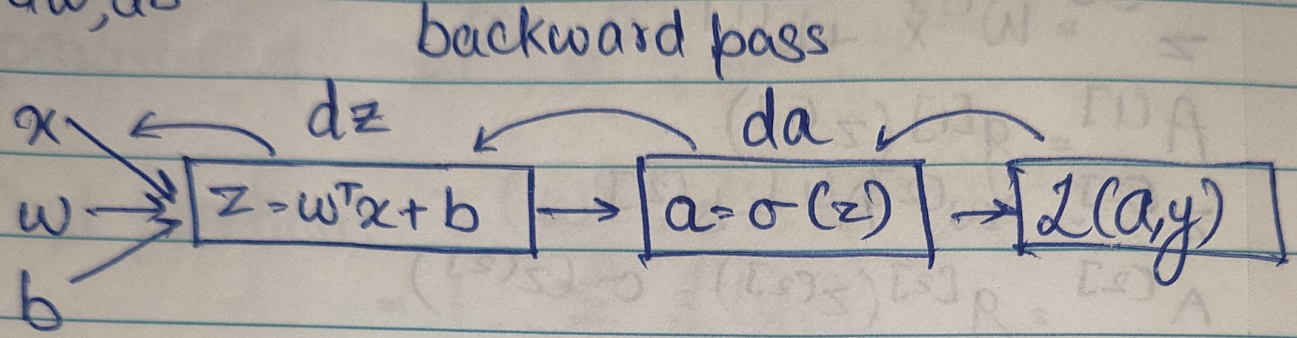
$$db^{[2]} = \frac{1}{m} np.sum(dz^{[2]}, axis=1, keepdims=True)$$

$$dz^{[1]} = \underbrace{W^{[2]T} dz^{[2]}}_{(n^{[1]}, m)} \times \underbrace{g^{[1]'}(z^{[1]})}_{\text{elementwise product}}$$

$$db^{[1]} = \frac{1}{m} np.sum(dz^{[1]}, axis=1, keepdims=True)$$

Backpropagation Intuition

dw, db



Forward Pass

$$da = \frac{d(\mathcal{L}(a,y))}{da} = -y\log a - (1-y)\log(1-a)$$

$$= \frac{-y}{a} + \frac{1-y}{1-a}$$

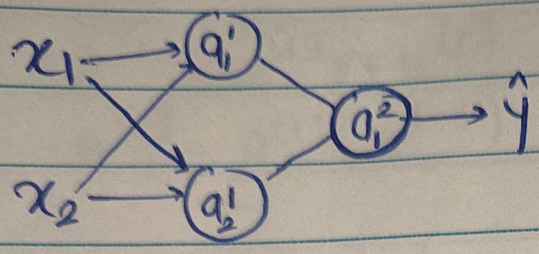
$$dz = da \cdot g'(z) = \left(\frac{-y}{a} + \frac{1-y}{1-a}\right)(a(1-a))$$

$$\left(\frac{dL}{da}\frac{da}{dz}\right)$$

$$dw = x\,dz$$

$$db = dz$$

Random initialization



if $w, b \rightarrow 0$ matrices

$a_1^{[1]} = a_2^{[1]} \longrightarrow dz_1^{[1]} = dz_2^{[1]}$

Both hidden units have same influence on next layer - computing same function

∴ we init. randomly

$W^{[1]} = np.random.randn((2,2)) * 0.01$

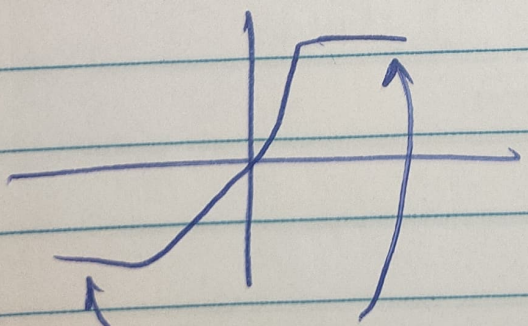
$b^{[1]} = np.zero((2,1))$

$W^{[2]} = \ldots$

$b^{[2]} = 0$

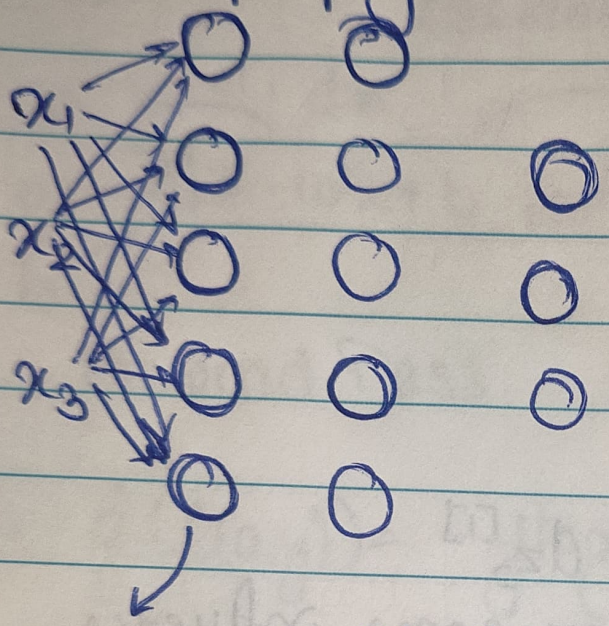
prefer to init to small vals.

because if w is big, z is big ∴ a may be very large or v. small



slow slope ∴ slow training

Deep L-Layer Neural Network
forward propagation



$X: z^{[1]} = w^{[1]} a^{[0]} + b^{[1]}$

$a^{[1]} = g^{[1]}(z^{[1]})$

$z^{[2]} = w^{[2]} a^{[1]} + b^{[2]}$

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

$\vdots$

$z^{[l]} = w^{[l]} a^{[l-1]} + b^{[l]}$

$a^{[l]} = g^{[l]}(z^{[l]})$

Vectorized version

$$z^{[1]} = w^{[1]}x + b^{[1]} \qquad z^{[2]} = w^{[2]}a^{[1]} + b^{[2]}$$

$$A^{[1]} = g^{[1]}(z^{[1]}) \qquad A^{[2]} = g^{[2]}(z^{[2]})$$

$$Z = \begin{bmatrix} | & | & & | \\ z^{[1]} & z^{[2]} & \cdots & z^{[m]} \\ | & | & & | \end{bmatrix}$$

$$A = \begin{bmatrix} | & | & & | \\ a^{[1]} & a^{[2]} & \cdots & a^{[m]} \\ | & | & & | \end{bmatrix}$$

explicit for loop required for 1 to 1

$n^{[l]}$ (no of hidden units in layer l)

matrix dimentions

$l = 5$ (5 layer NN)

$z^{[1]} = w^{[1]} \cdot x + b^{[1]}$ → bias

$n^{[1]} = 3,\ n^{[2]} = 5,\ n^{[3]} = 4,\ n^{[4]} = 2,\ n^{[5]} = 1$

↳ activation vector for [1]

∴ dimension: (3,1) $(n^{[1]}, 1)$

$x \rightarrow$ 2 input features

∴ (2,1) dim $[n^0, 1]$

$\therefore \begin{bmatrix} \vdots \\ \vdots \end{bmatrix} = w^T \begin{bmatrix} \vdots \\ \vdots \end{bmatrix}$

(3×2) dim

$w^{[1]} = n^{[1]}, n^{[0]}$

$w^{[2]} = n^{[2]}, n^{[1]} = (5,3)$ dim.

$w^{[l]} = (n^{[l]}, n^{[l-1]})$ dim

$b^{[l]} = (n^{[l]}, 1)$ dim

same as w

same as b

$dw^{[l]} = (n^{[l]}, n^{[l-1]})$ $db^{[l]} = (n^{[l]}, 1)$

Vectorized Implementation

$Z^{[1]} = W^{[1]} \cdot X + b^{[1]}$

$[n', n^{\circ}]$ $[n^{[0]}, m]$ (stacked)

$[n^{[1]}, m]$ (stacked as $\begin{bmatrix} | & | \\ z^{[1]} & z^{[2]} \\ | & | \end{bmatrix}$

$Z^{[l]}, A^{[l]} : [n^{[l]}, m]$ dim

$dz^{[l]}, dA^{[l]} :$