Lab 1

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

$$g'(z) = \frac{-1}{(1+e^{-z})^2} \cdot (-e^{-z})$$

$$\frac{e^{-z}}{(1+e^{-z})^2} = \frac{1+e^{-z}-1}{(1+e^{-z})^2} = \frac{1}{1+e^{-z}} - \frac{1}{(1+e^{-z})^2}$$

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

$$g'(z) = g(z)\left[1 - g(z)\right]$$

b) $\tanh = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$

$$\times \frac{\cancel{e^{x}+(e^{x}-e^{-x})}}{\cancel{(e^{x}-e^{-x})}}$$

$$\cancel{\tanh = (e^{x}-e^{-x})^2}$$

$$\tanh'(x) = \frac{(e^{x}+e^{-x})\frac{d(e^{x}-e^{-x})}{dx} - (e^{x}-e^{-x})\frac{d(e^{x}+e^{-x})}{dx}}{(e^{x}+e^{-x})^2}$$

$$= \frac{(e^{x}+e^{-x})\left[e^{x}+e^{-x}\right] - (e^{x}-e^{-x})(e^{x}-e^{-x})}{(e^{x}+e^{-x})^2}$$

$$= \frac{(e^{x}+e^{-x})^2 - (e^{x}-e^{-x})^2}{(e^{x}+e^{-x})^2}$$

$= 1 - \tanh^2$

$= 1 - [\tanh(x)]^2$

c) ReLU function

$\mathrm{ReLU}(x) = \max(0, x)$

$\mathrm{ReLU}'(x) = \begin{cases} 0 & \text{if } x \le 0 \\ 1 & \text{if } x > 0 \end{cases}$

d) Leaky ReLU

$\text{Leaky ReLU} \begin{cases} x & \text{if } x > 0 \\ c * x & \text{if } x \le 0 \end{cases}$

$\mathrm{der}(c\mathrm{R}) \begin{cases} 1 & \text{if } x > 0 \\ c & \text{if } x \le 0 \end{cases}$

e) Softmax

$$\mathrm{soft}(x) = \frac{e^{x_i}}{\sum_{i=1}^{k} e^{x_i}} = s(x)$$

$$J = \begin{bmatrix} \frac{\partial s_1}{\partial x_1} & \frac{\partial s_2}{\partial x_1} & \frac{\partial s_3}{\partial x_1} & \cdots & \frac{\partial s_k}{\partial x_1} \\ \vdots & & & & \\ \frac{\partial s_k}{\partial x_1} & & & & \frac{\partial s_k}{\partial x_k} \end{bmatrix}_{k \times k}$$

Numerical for Softmax:

$$x = \begin{bmatrix} 2 \\ 1 \\ 0.1 \end{bmatrix} \qquad s(x) = \begin{bmatrix} \frac{e^2}{e^2 + e^1 + e^{0.1}} \\ \frac{e^1}{e^2 + e^1 + e^{0.1}} \\ \frac{e^{0.1}}{e^2 + e^1 + e^{0.1}} \end{bmatrix} = \begin{bmatrix} 0.66 \\ 0.25 \\ 0.09 \end{bmatrix}$$

for diagonal elements. (i.e) $i=j$

$$\frac{\partial S_i}{\partial x_i} = \frac{\partial \cdot \frac{e^{x_i}}{e^{x_1}+e^{x_2}+\dots e^{x_k}}}{\partial x_i}$$

$$= \frac{(e^{x_1}+e^{x_2}\dots\dots+e^{x_k})e^{x_i} - e^{x_i}(1)e^{x_i}}{(e^{x_1}+e^{x_2}+\dots\dots+e^{x_k})^2}$$

$$= \frac{e^{x_i}}{(e^{x_1}+e^{x_2}+\dots\dots+e^{x_k})}\left[\frac{1}{\cancel{e^{x_1}+e^{x_2}+\dots+e^{x_k}}} - \frac{e^{x_i}}{e^{x_1}+\dots+e^{x_k}}\right]$$

$$\frac{\partial S_i}{\partial x_i} = \cancel{S_i(x)}\ S(x_i)\left[1 - S(x_i)\right]$$

for off diagonal.

$$\frac{\partial S_i}{\partial x_j} = \frac{\partial \frac{e^{x_i}}{e^{x_1}+e^{x_2}+\dots+e^{x_k}}}{\partial x_j}$$

$$= \frac{(e^{x_1}+\dots\dots+e^{x_k})\cancel{e^{x_i}}\ e^{x_i}\frac{\partial x_i}{\partial x_j} - e^{x_i}e^{x_j}}{(e^{x_1}+e^{x_2}+\dots\dots+e^{x_k})^2}$$

$$= \frac{0 - e^{x_i}e^{x_j}}{(e^{x_1}+e^{x_2}+\dots+e^{x_k})^2}$$

$$\frac{\partial S_i}{\partial x_j} = -S_i S_j$$

tanh = 2sigmoid(2x) - 1

$$2\sigma(2x) - 1 = \frac{2}{1+e^{-2x}} - 1$$

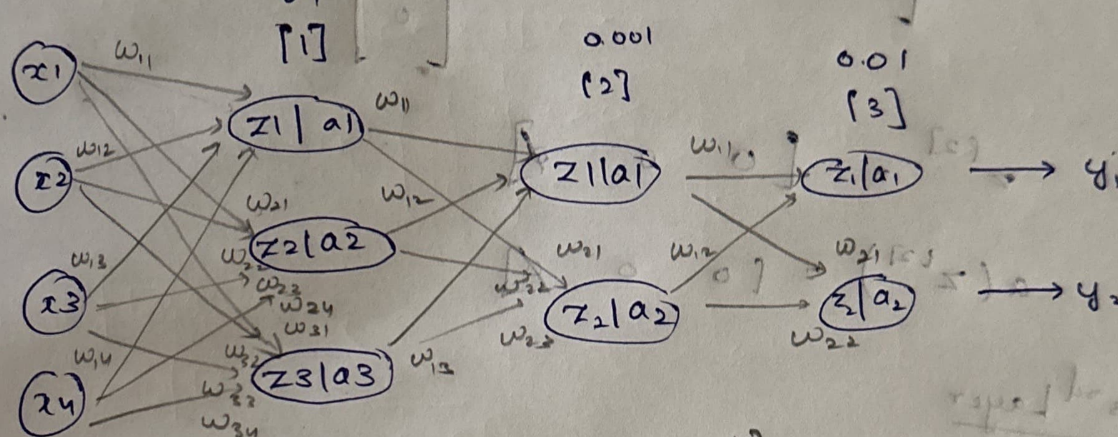
$$= \frac{2-1-e^{-2x}}{1+e^{-2x}}$$

$$= \frac{1-e^{-2x}}{1+e^{-2x}} = \tanh(x)$$

Lab 2

Hidden layers - ReLU

output layer - softmax.



$$W^{[3]} = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix}$$

$$W^{[2]} = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{bmatrix}$$

$$W^{[1]} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{22} & w_{24} \\ w_{21} & w_{32} & w_{23} & w_{34} \end{bmatrix}$$

$$X = \begin{bmatrix} -2.4 \\ 1.2 \\ -0.8 \\ 1.1 \end{bmatrix}$$

$$z^{[1]} = Wx + b \rightarrow 0.$$

$$z^{[1]} = \underbrace{\begin{bmatrix} 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 \end{bmatrix}}_{3\times4} \underbrace{\begin{bmatrix} -2.4 \\ 1.2 \\ -0.8 \\ 1.1 \end{bmatrix}}_{4\times1}$$

$$z^{[1]} = \begin{bmatrix} -0.09 \\ -0.09 \\ -0.09 \end{bmatrix}$$

$$a(z^{[1]}) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

2nd Layer

$$\begin{bmatrix} 0.001 & 0.001 & 0.001 \\ 0.001 & 0.001 & 0.001 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$z^{[2]} \ [0 \quad 0]$$

$$a(z^{[2]}) = [0 \quad 0]$$

3rd Layer

$$\begin{bmatrix} 0.01 & 0.01 \\ 0.01 & 0.01 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$z^{[3]} : [0 \quad 0]$$

$$a(z^{[3]}) : [0 \quad 0]$$

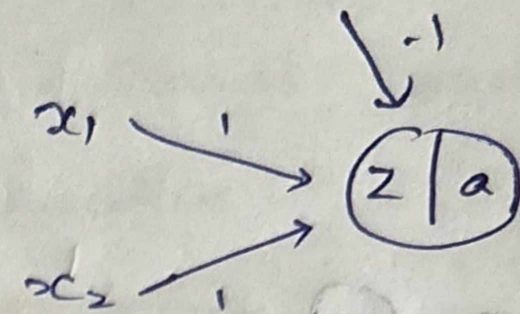
$\downarrow \quad \downarrow$ softmax

$0.5 \quad 0.5$

Proof

AND function

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1



$\boxed{x_1 + x_2 - 1 = z.}$

perceptron

x_1	x_2	z	a
0	0	-1	0
1	0	0	0
0	1	0	0
1	1	1	1

It can model an AND gate

$\boxed{z = x_1 + x_2}$

bias term = 0

OR function

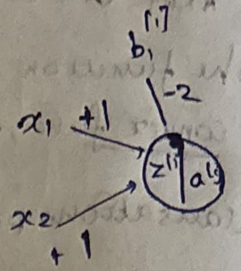
x_1	x_2	y
0	0	0
1	0	1
0	1	1
1	1	1

perceptron

x_1	x_2	z	a
0	0	0	0
1	0	1	1
0	1	1	1
1	1	2	1

XOR function

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0



$$x_1 + x_2 - 2 = z$$

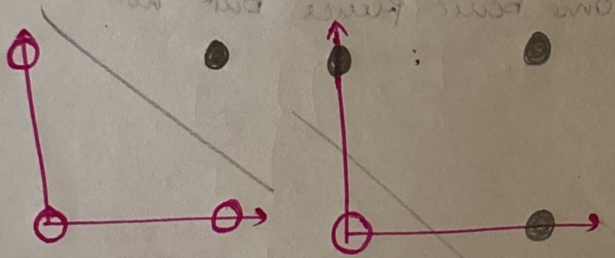
x_1	x_2	z_1	a
0	0	-2	0
0	1	-1	0
1	0	-1	0
1	1	0	0

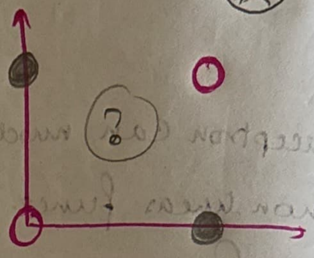
Perceptron can model only linear funcs

cannot model XOR. Why?

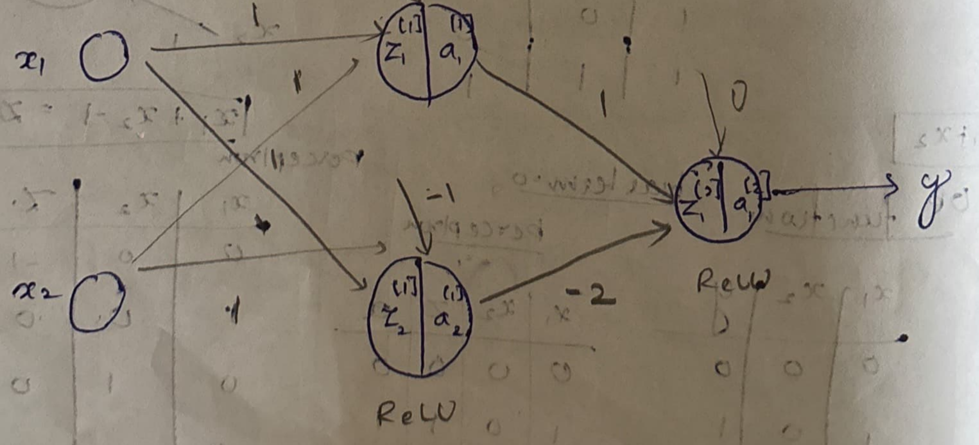
inherently non linear function

~~so~~ cannot be seperated by linear boundary





↳ cannot draw a linear decision boundary



$ReLU(x) : max(0, x)$

$z_1^{[1]} = x_1 + x_2 + 0 \qquad z_2^{[1]} = x_1 + x_2 - 1$

x_1	x_2	$z_1^{[1]}$	$a_1^{[1]}$
0	0	0	0
0	1	1	1
1	0	1	1
1	1	2	2

x_1	x_2	$z_2^{[1]}$	$a_2^{[1]}$
0	0	-1	0
0	1	0	0
1	0	0	0
1	1	1	1

$--\rightarrow a = \begin{bmatrix} a_1^{[1]} \\ a_2^{[1]} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ or $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ or $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ or $\begin{bmatrix} 2 \\ 1 \end{bmatrix}$

$z_1^{[2]} = a_1^{[1]} - 2a_2^{[1]}$

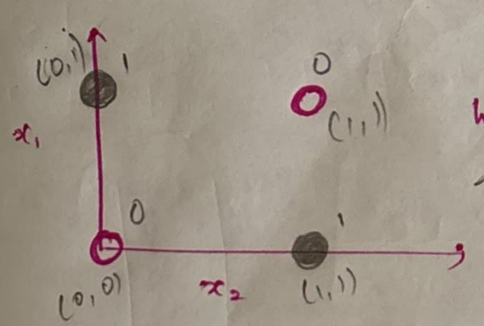
$a_1^{[1]}$	$a_2^{[1]}$	$z_1^{[2]}$	$a_1^{[2]}$
0	0	0	0 ✓
1	0	1	1 ✓
1	0	1	1 ✓
2	1	0	0 ✓

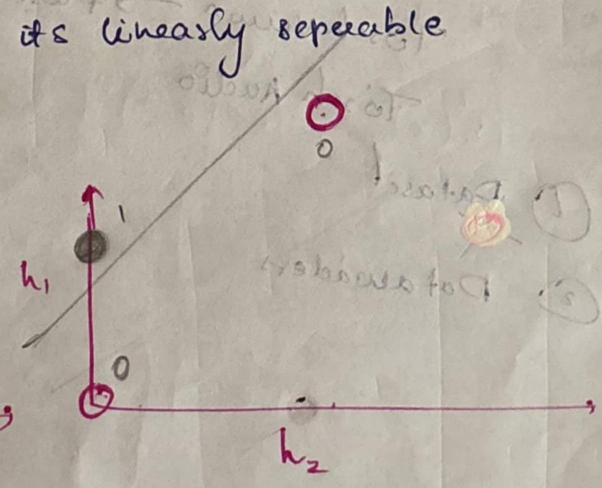
works ✓

note :- introducing additional layers can model nonlinear fn

how does it work?

input data (x_1, x_2) is transformed from original space to new space where its linearly seperable (h_1, h_2)





weights & bias will be learnt from Data (X)