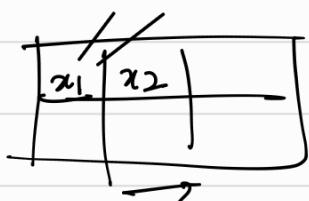
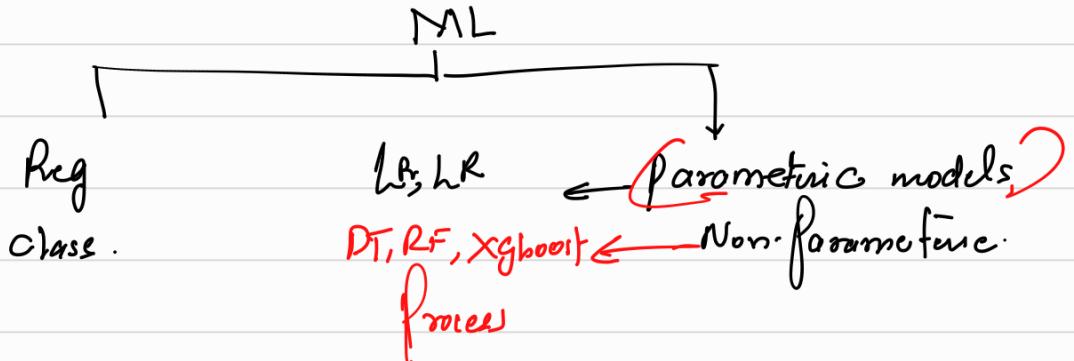


Deep learning

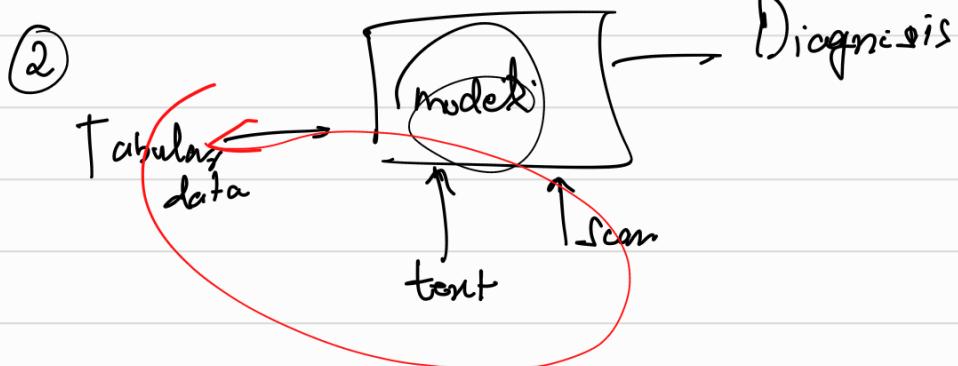
$$y = f(x) + \epsilon$$

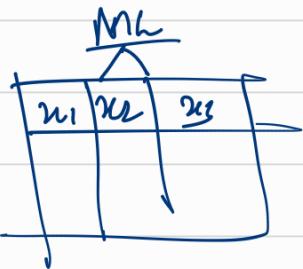
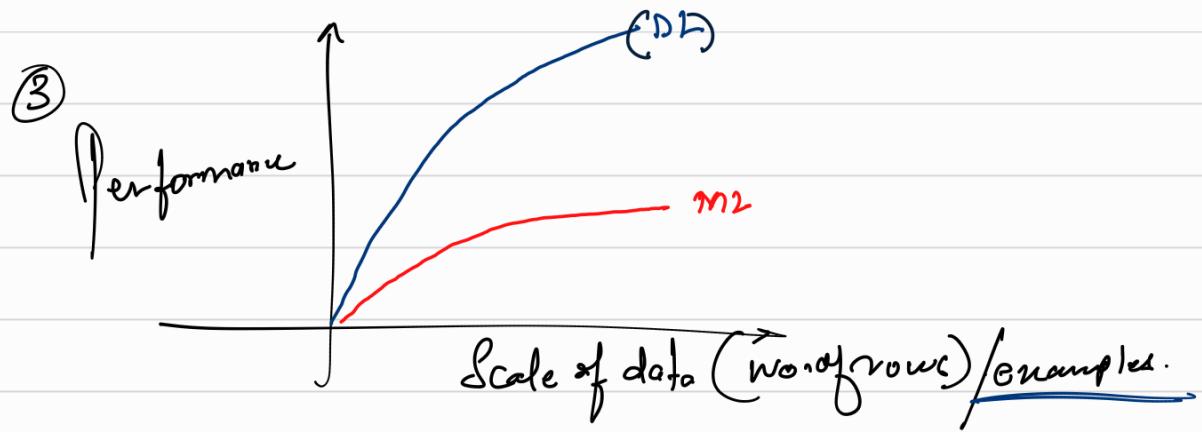


$$y \sim f(x)$$

① $x \rightarrow$ (video, audio, text, tabular data, graph).

multimodality



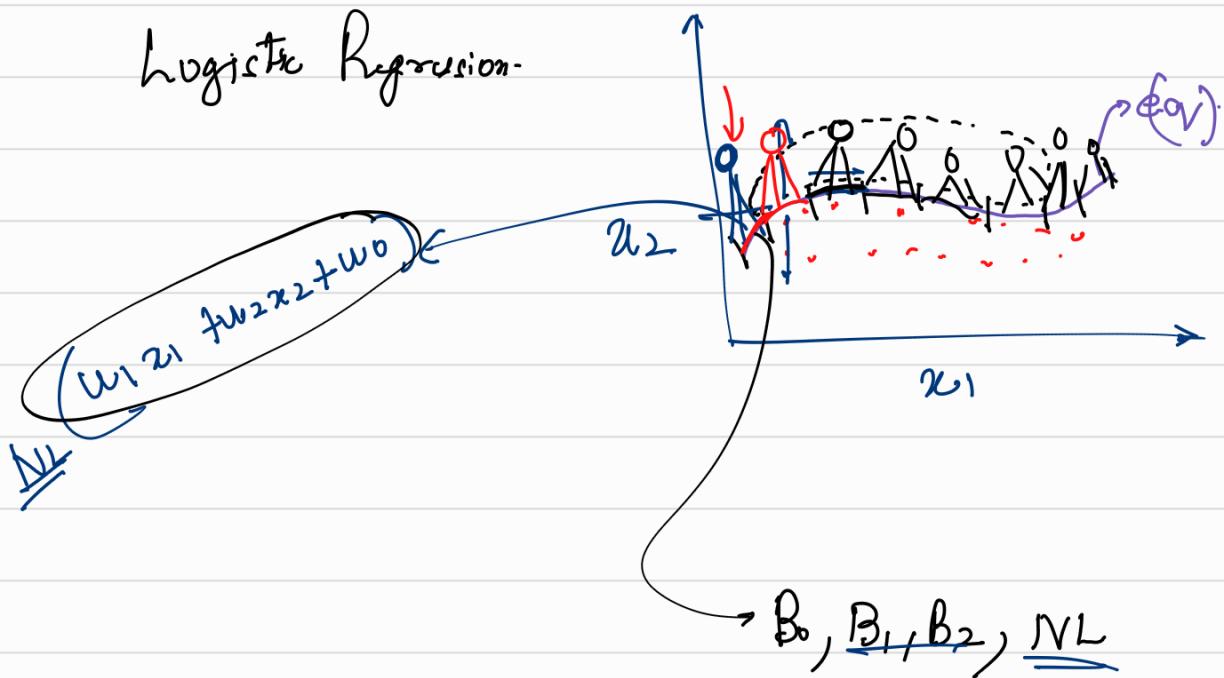


$D_h \rightarrow$ automated feature engineering

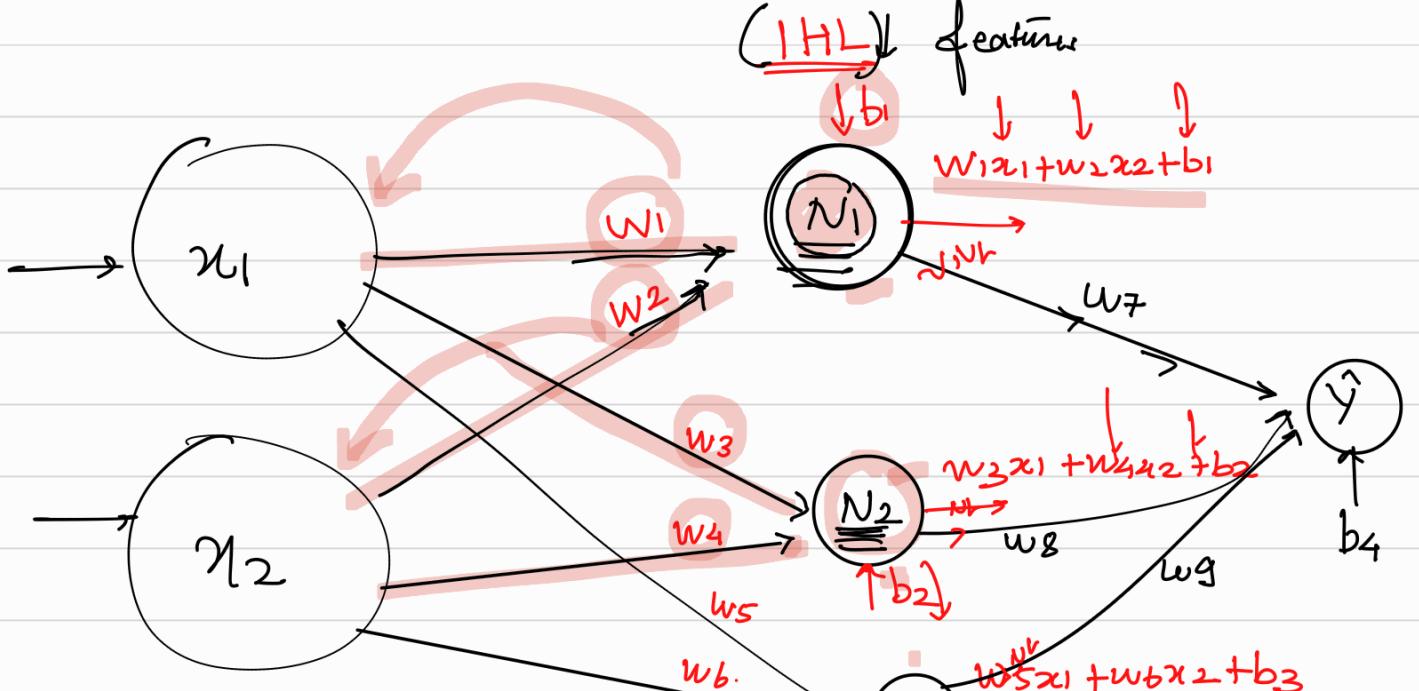


What is a Neuron

Logistic Regression-



m_1	m_2	y



$$N_1 = \underbrace{(w_1 x_1 + w_2 x_2 + b_1)}_{\text{feature}}, \quad N_2 = \underbrace{(w_3 x_1 + w_4 x_2 + b_2)}_{\text{feature}}, \quad N_3 = \underbrace{(w_5 x_1 + w_6 x_2 + b_3)}_{\text{feature}}$$

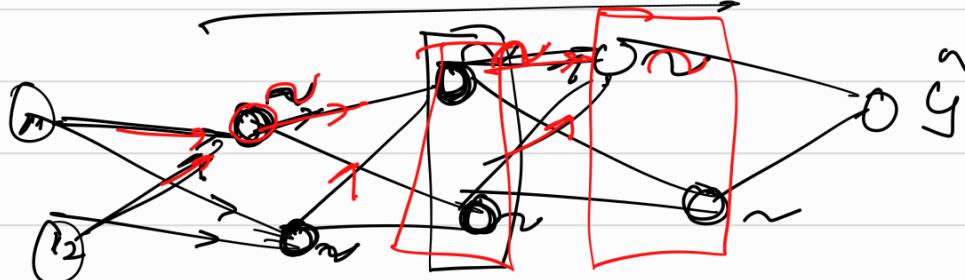
$$\hat{y} = w_7 N_1 + w_8 N_2 + w_9 N_3 + b_4$$

$$\Rightarrow \hat{y} = \underbrace{w_7}_{\text{feature}} \underbrace{(w_1 x_1 + w_2 x_2 + b_1)}_{\text{feature}} + \underbrace{w_8}_{\text{feature}} \underbrace{(w_3 x_1 + w_4 x_2 + b_2)}_{\text{feature}} + \underbrace{w_9}_{\text{feature}} \underbrace{(w_5 x_1 + w_6 x_2 + b_3)}_{\text{feature}} + b_4$$

$$\Rightarrow \hat{y} = x_1 (w_7 w_1 + w_8 w_3 + w_9 w_5) + x_2 (w_7 w_2 + w_8 w_4 + w_9 w_6)$$

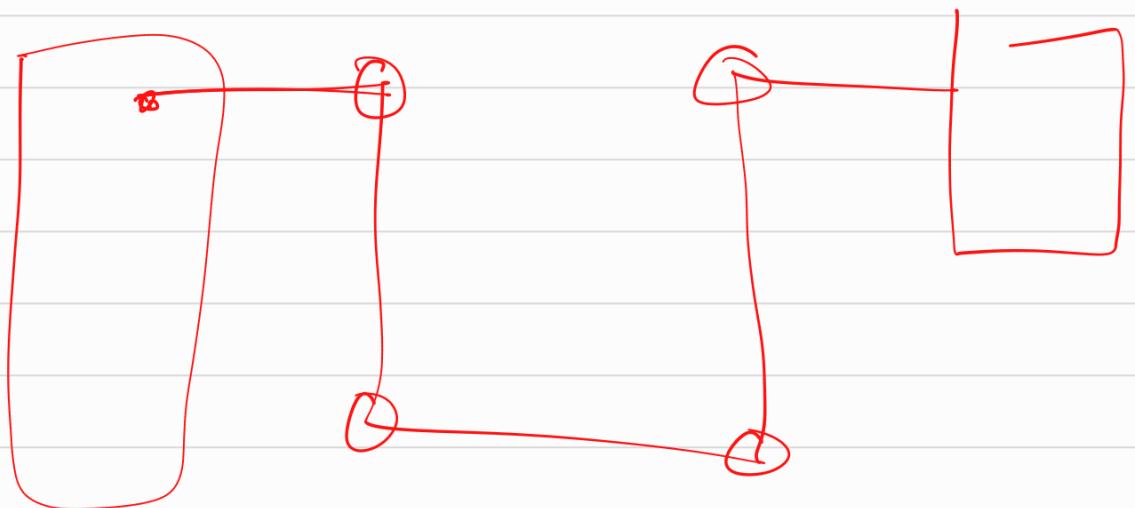
$$+ w_7 b_1 + w_8 b_2 + w_9 b_3 + b_4$$

$$\Rightarrow \hat{y} = A x_1 + B x_2 + C$$



1 H_L

→ nth H_L

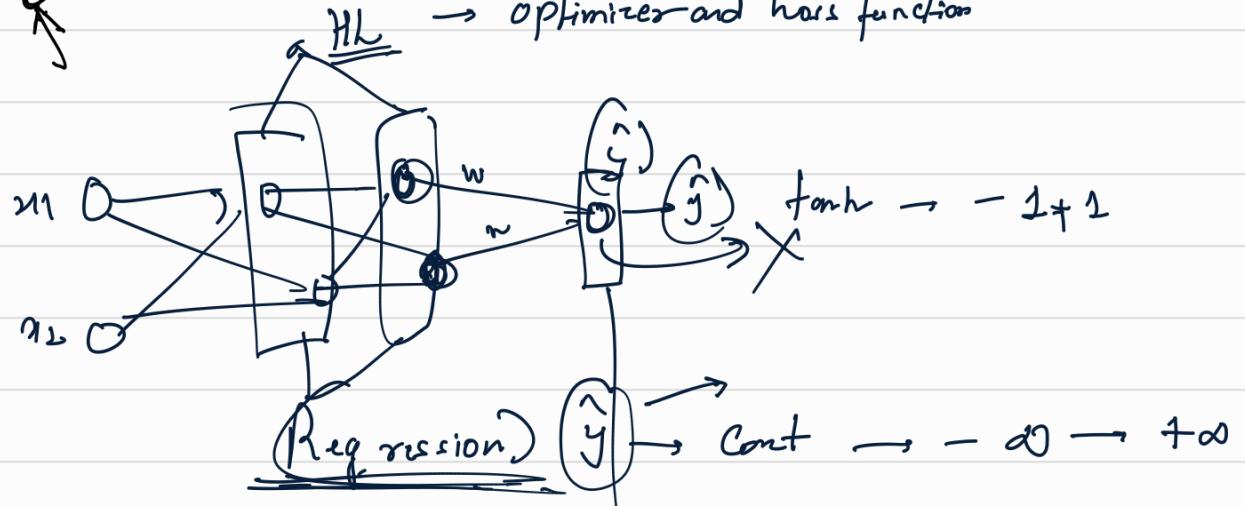
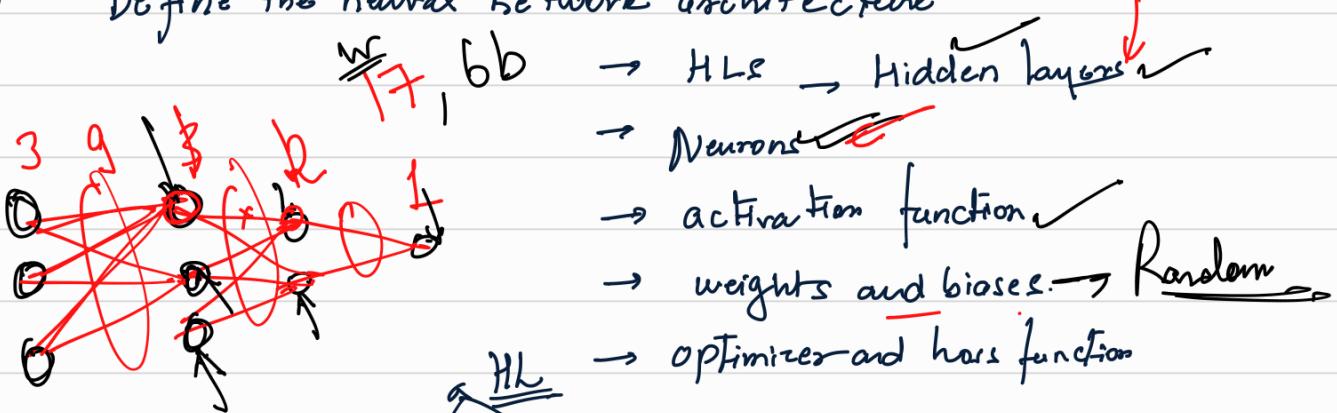


Gradient Descent

: minimise the loss by updating the W, B .
for all neurons.

RanCham

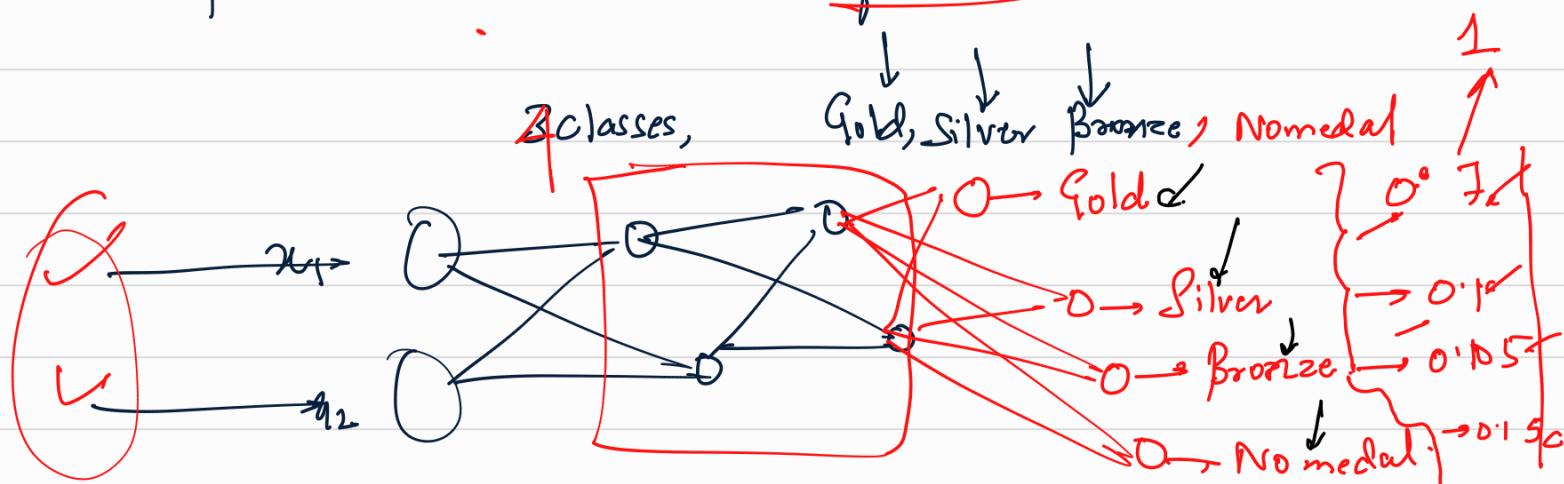
1) Define the neural network architecture

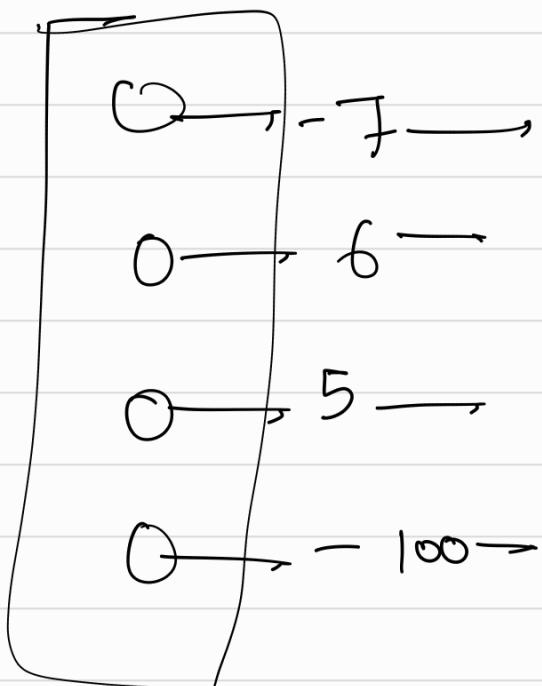
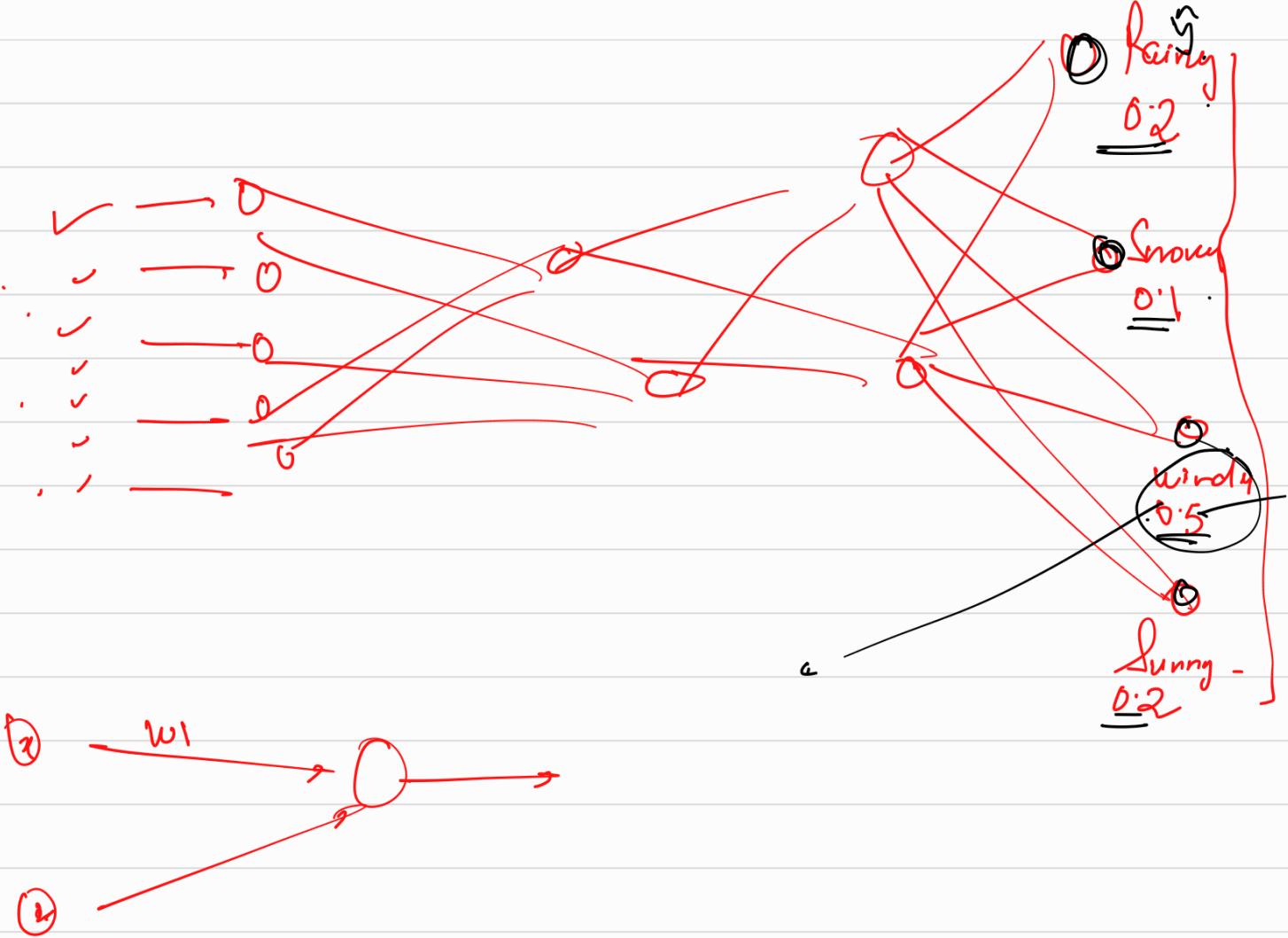


Classification (Binary). $\rightarrow \text{Sig}(y)$ \rightarrow 0-1

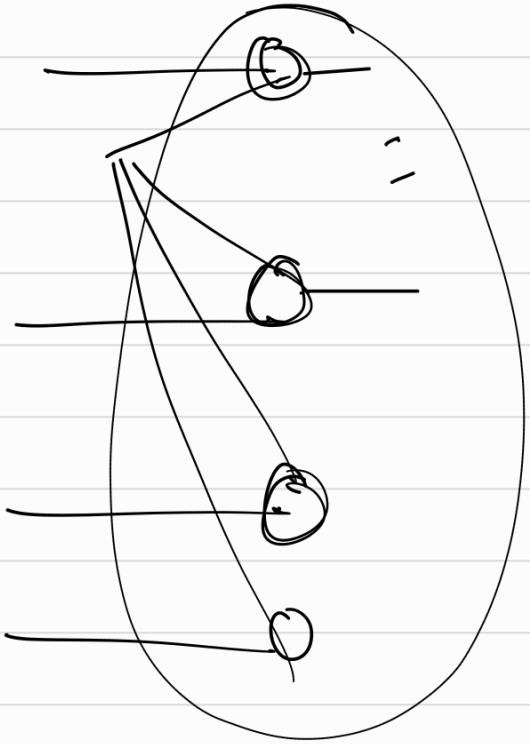
Classification (multiclass).

Softmax (y).





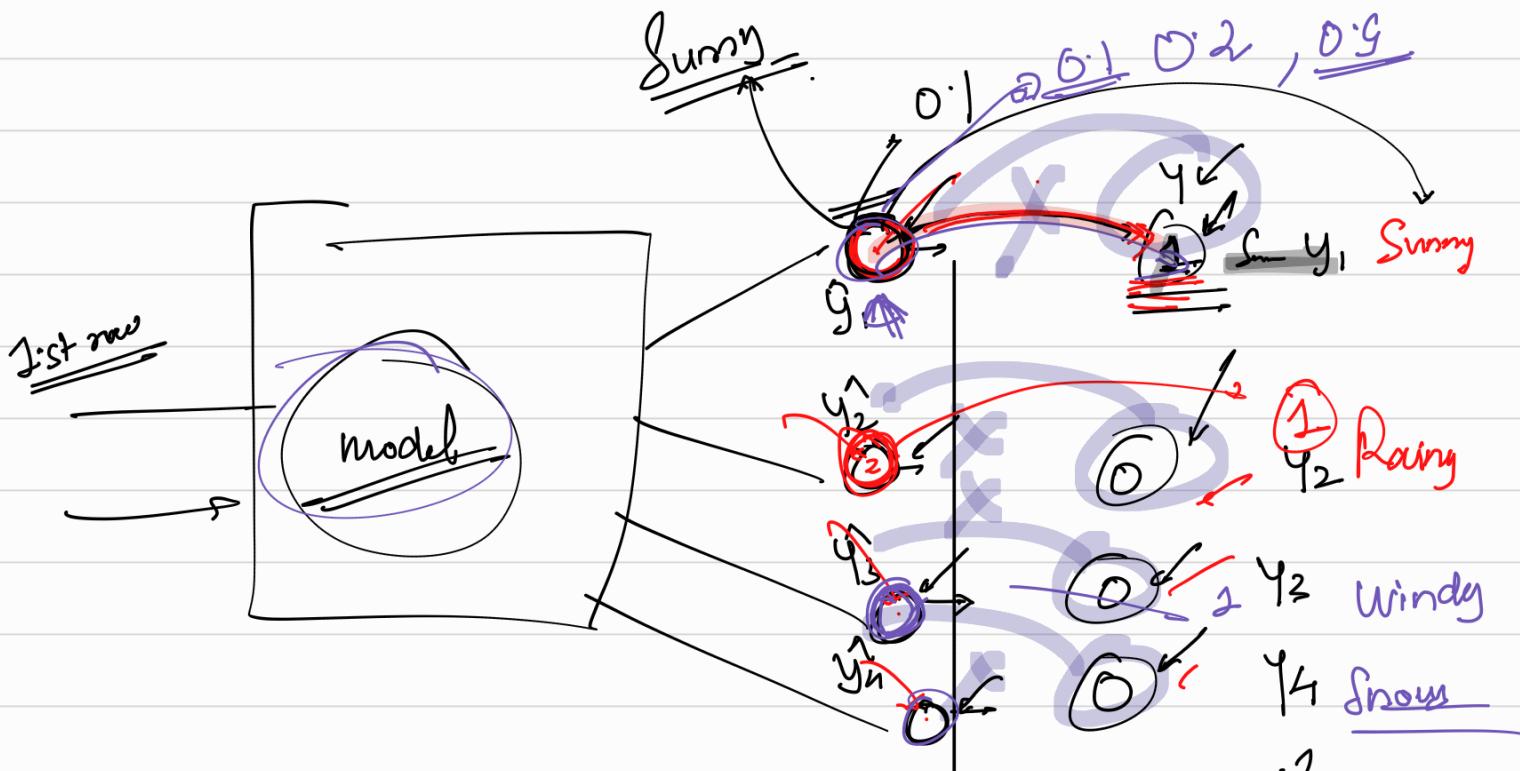
x_1	x_2	Weather (y)	$y-Su$	$y-Wi$	$y-Ra$	w_i	y
		Sunny	1	0	0	0	0
		Rainy	0	1	0	0	0
		Windy	0	0	1	1	0
		Sunny	1	0	0	0	0
		Shady	0	0	0	0	1



0

0

0
0



log loss = for all rows of data sum - $(y_1 \log(\underline{y_1}) + y_2 \log(\underline{y_2}) + y_3 \log(\underline{y_3}) + y_4 \log(\underline{y_4}))$

$$- (\log 0.2)$$

$$= \underline{0.69}$$

$$- (\log 0.2)$$

$$= \underline{0.09}$$

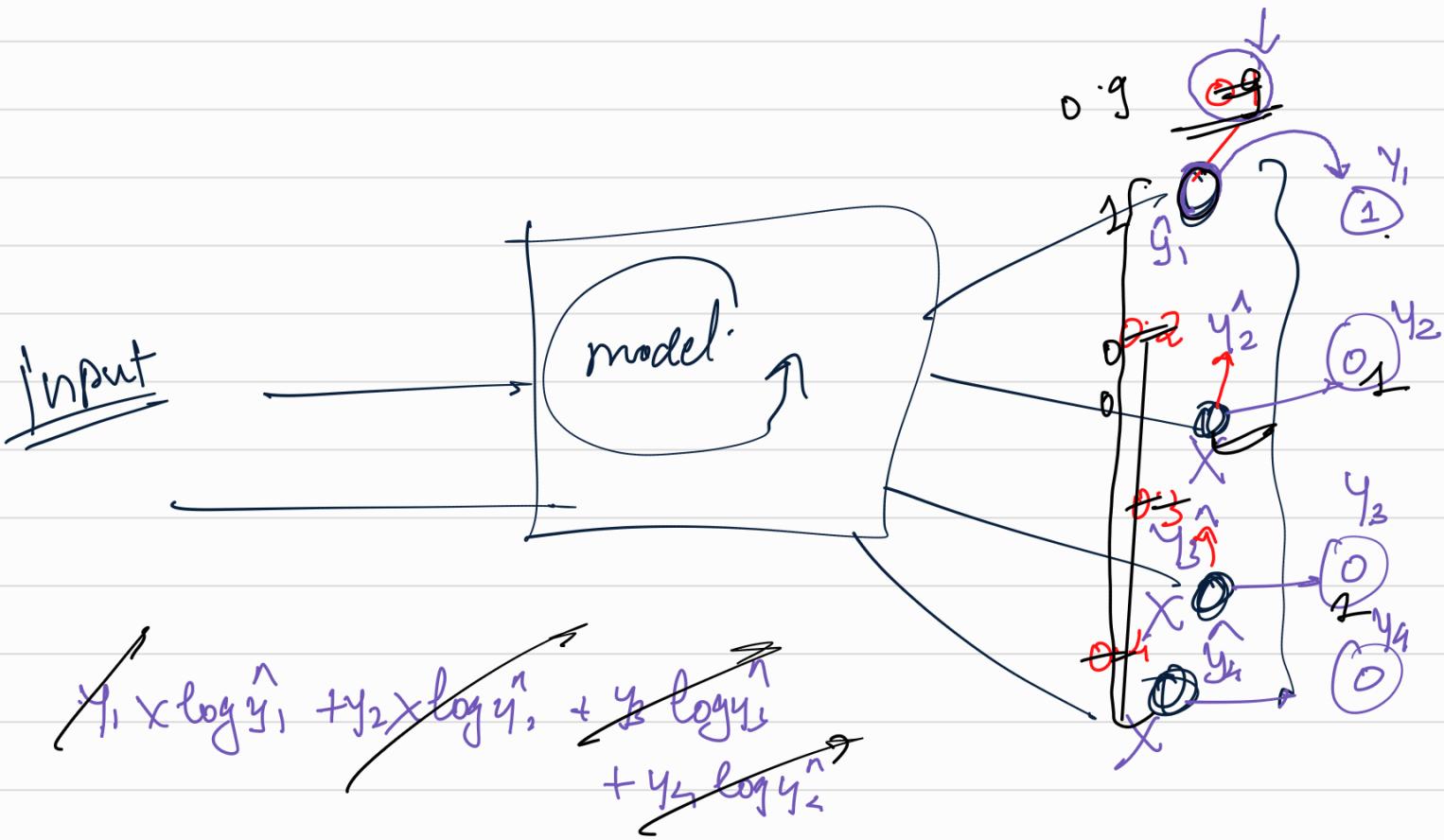
$$- \underline{\log(0.95)} = 0.02$$

$$\log 1 = 0$$

$$y_1 \log \hat{y}_1 + y_2 \log \hat{y}_2 + y_3 \log \hat{y}_3 + y_4 \log \hat{y}_4$$

~~$y_1 \log y_1 + y_2 \log y_2 + y_3 \log y_3 + y_4 \log y_4$~~

$$\log(0.1) = -1$$



$$-\left(\log \hat{y}_i\right)$$

$$\cancel{\log 1 = 0}$$

$$-\left(\log 0.1\right)$$

↓

$$-\left(1\right)$$

VS

VS

$$-\cancel{\log 0.9}$$

↓

$$0.045$$

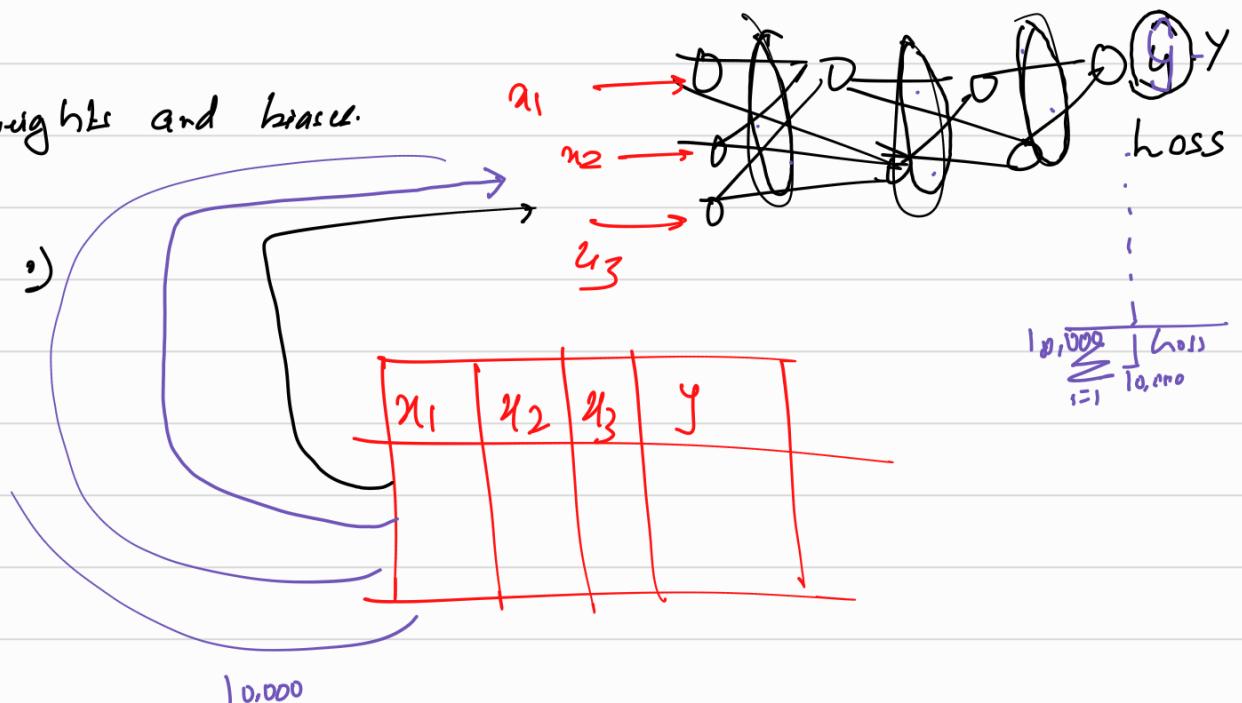
0
0
0

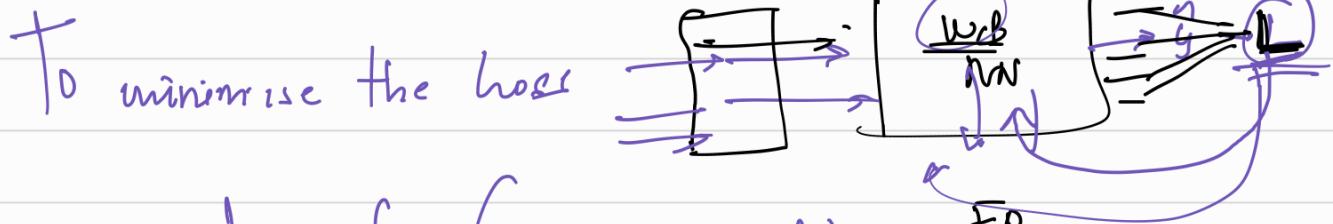
$$(0)$$

softmax, ohe

log loss

Random weights and biases.

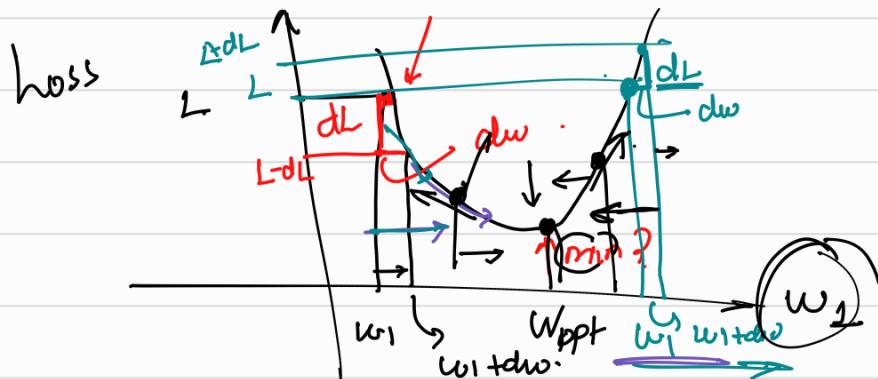
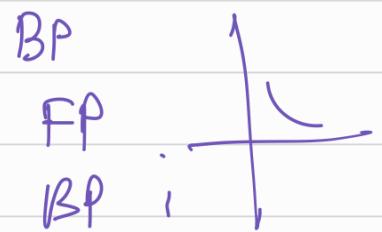




$$L = f(y_{\text{predicted}}, y_{\text{actual}})$$

$$L = f(\underline{w}, \underline{b}, \underline{y_{\text{actual}}})$$

$$L = f(w, b)$$



$$\frac{dw}{dw} = -\text{ve.}$$

which side works out +ve
more

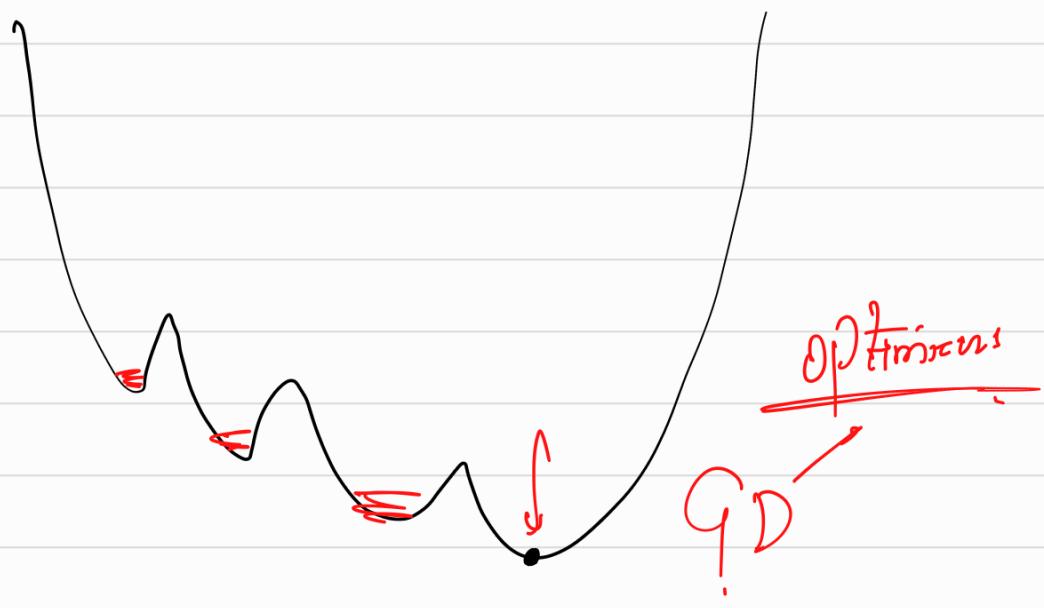
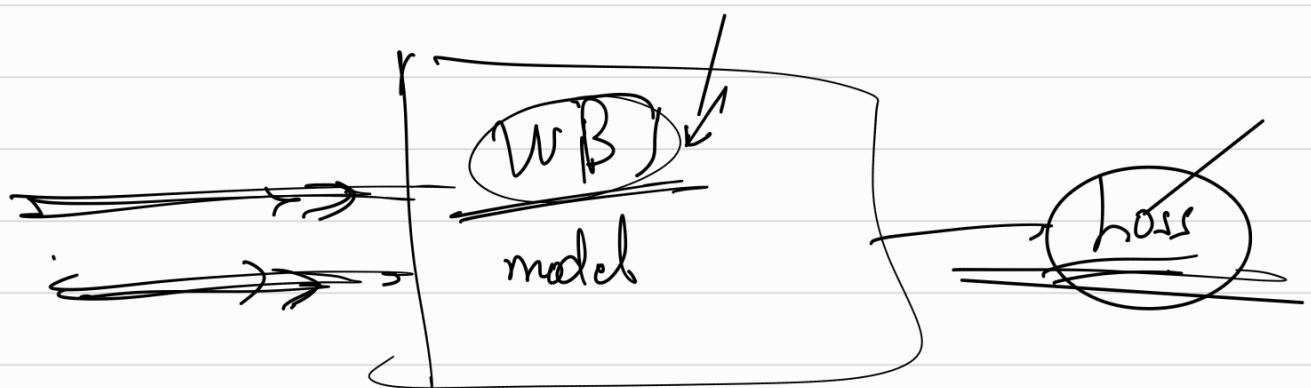
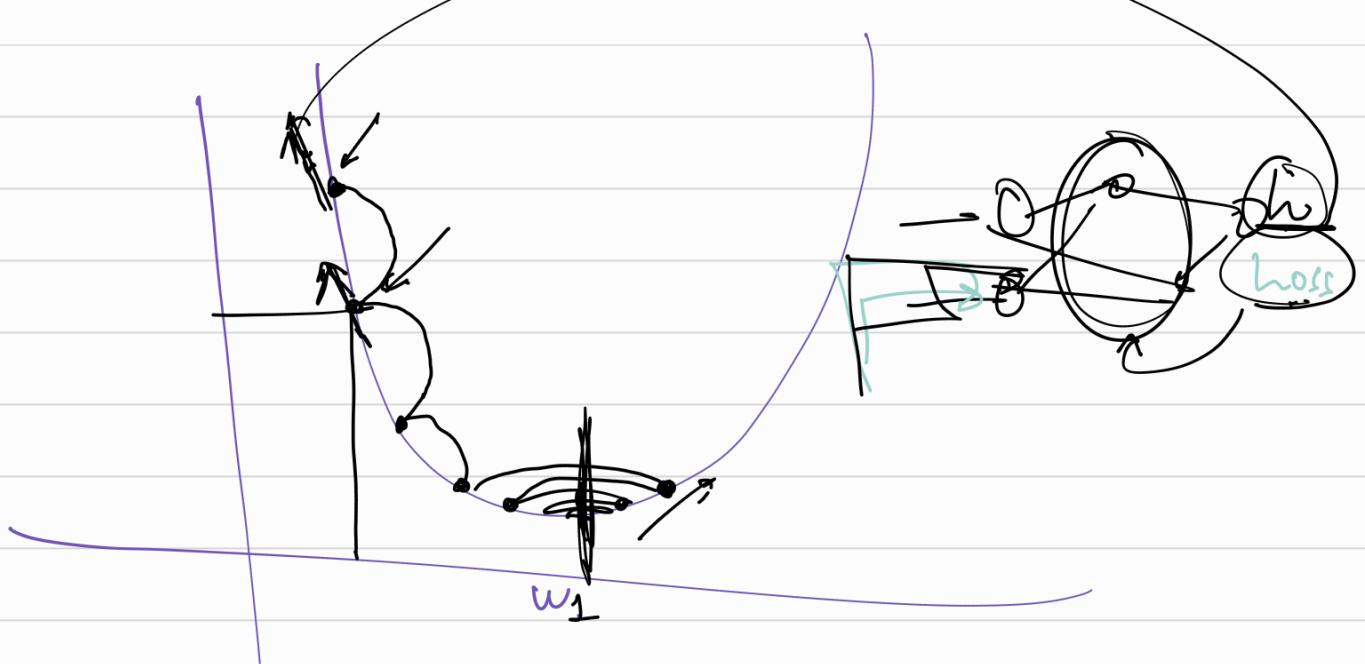
$$\frac{dw}{dw} = +\text{ve}$$

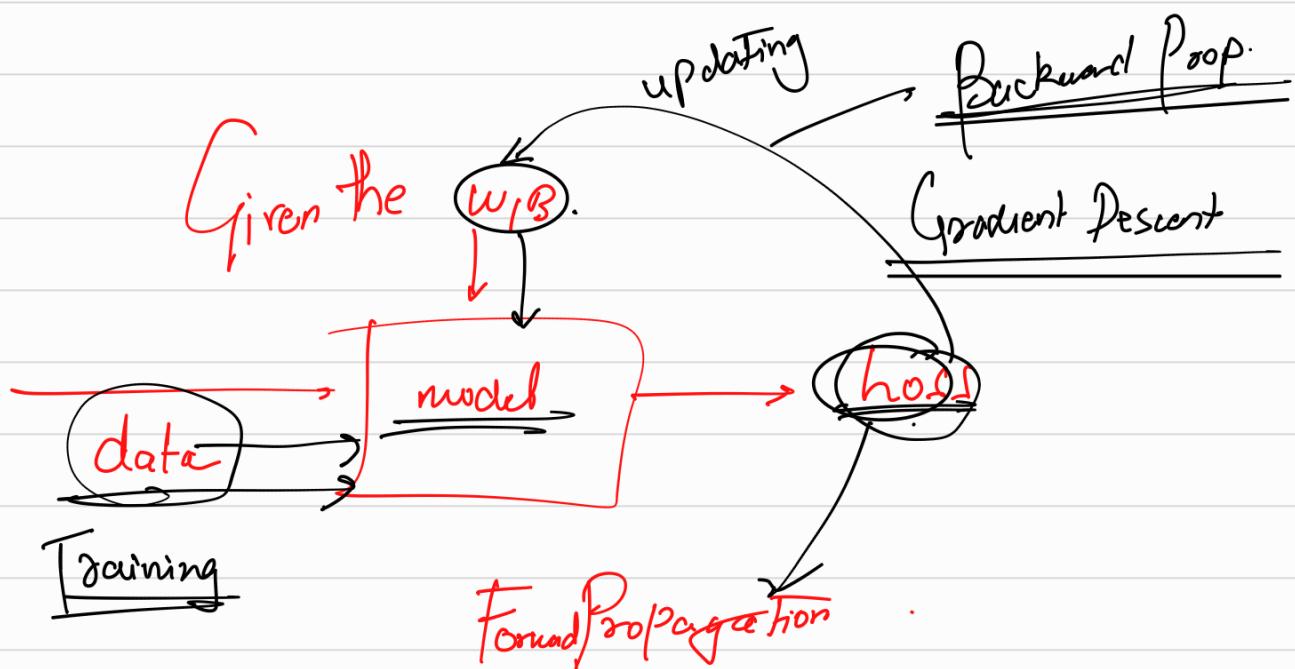
$$(w = -\text{ve.})$$

$$w_{\text{new}} = w_{\text{old}} - \alpha \left(\frac{dw}{dw} \right)$$

Diagram illustrating the update rule for weights. The new weight w_{new} is calculated as the old weight w_{old} minus the learning rate α times the partial derivative of the loss with respect to the weight w .

Applicable for all weights and biases.





$(\text{IFP} + \text{IBP}) \rightarrow \underline{\text{epoch}}$

$(\text{IFP}, \text{IBP}) \rightarrow \text{epoch}$ → $\downarrow \text{epoch}$

$\boxed{\text{IFP} + \text{IBP}}$

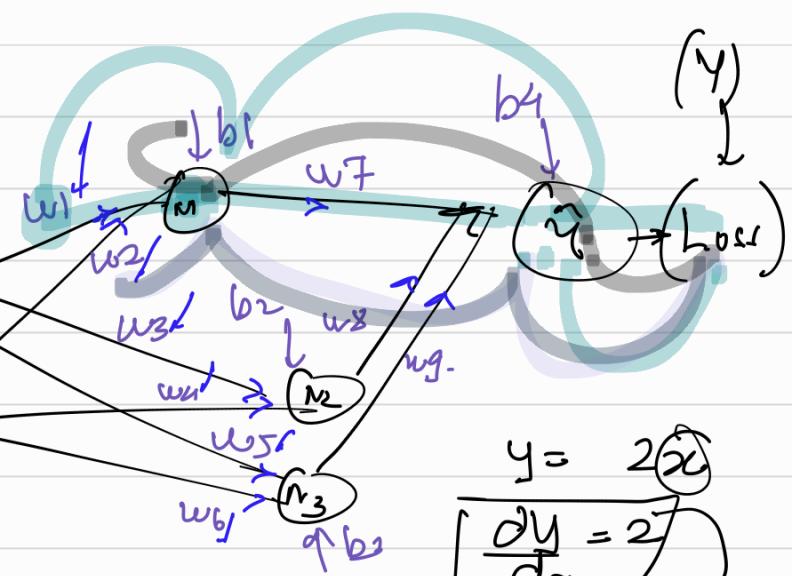
Lecture-3

$$N_1 = w_1 x_1 + w_2 x_2 + b_1 \quad (1)$$

$$N_2 = w_3 x_1 + w_4 x_2 + b_2$$

$$N_3 = w_5 x_1 + w_6 x_2 + b_3$$

$$\hat{y} = w_7 N_1 + w_8 N_2 + w_9 N_3 + b_4$$



$$y = 2x$$

$$\frac{\partial y}{\partial x} = 2$$

$$\frac{\partial L}{\partial \hat{y}} = \sum_{i=1}^n \frac{1}{n} (y_i - \hat{y}_i)^2 \rightarrow \text{MSE}$$

$$\frac{\partial L}{\partial w_7} = \frac{\partial L}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial w_7} \Rightarrow \sum_{i=1}^n \frac{1}{n} \times 2 \times \frac{(y_i - \hat{y}_i)}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial w_7} \times N_1$$

$$\frac{\partial h}{\partial w_9}, \frac{\partial h}{\partial w_1}$$

6 mins

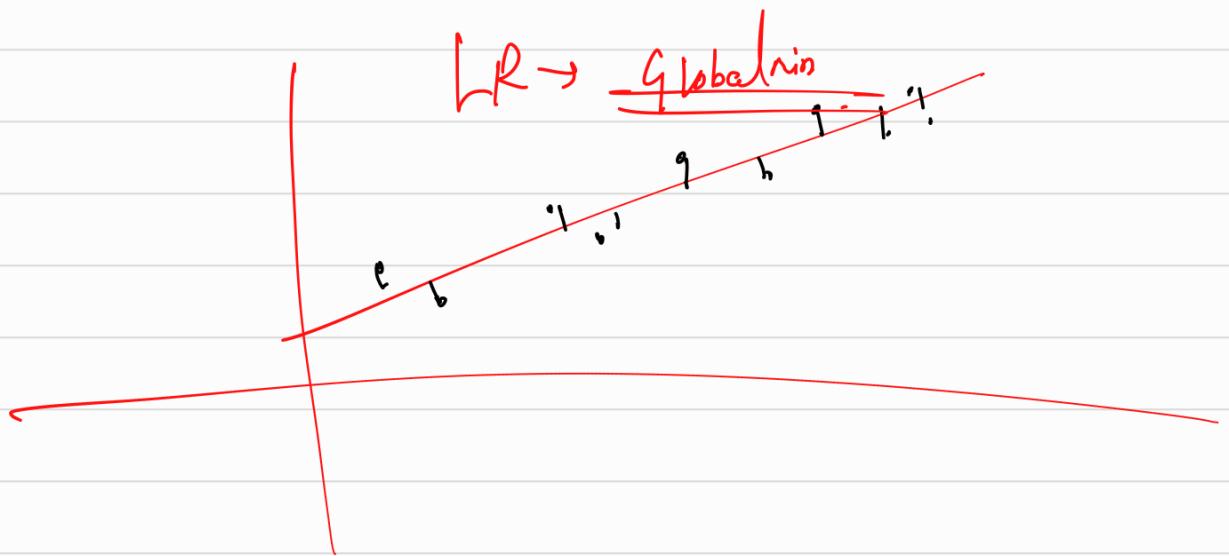
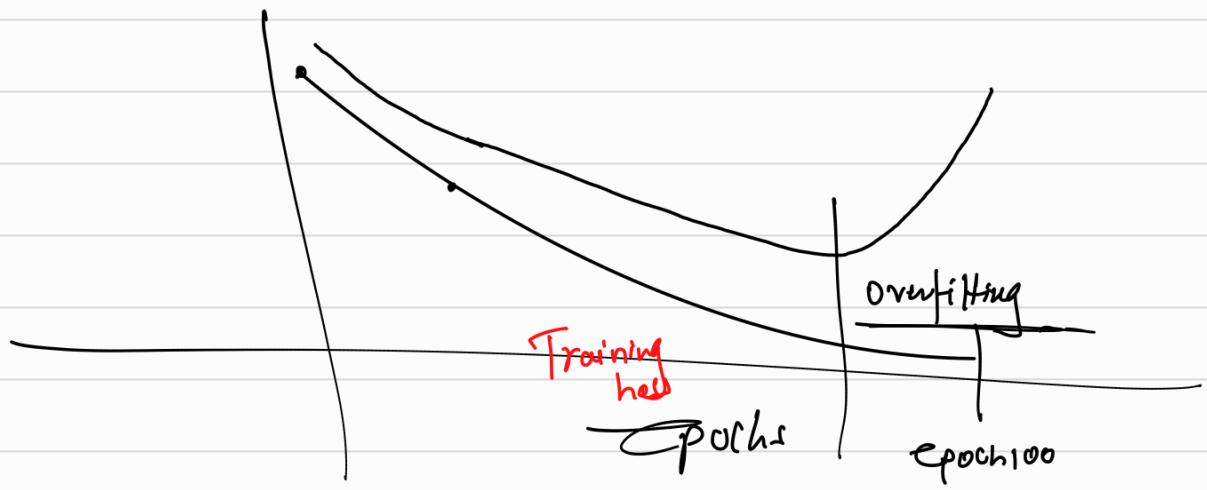
$$\frac{\partial L}{\partial w_9} = \left(\frac{\partial h}{\partial \hat{y}} \right) \times \frac{\partial \hat{y}}{\partial w_9} \rightarrow N_3$$

$$\frac{\partial h}{\partial w_1} = \frac{\partial h}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial N_1} \times \frac{\partial N_1}{\partial w_1}$$

$$\hat{y} = w_7 N_1 + w_8 N_2 + w_9 N_3 + b_4$$

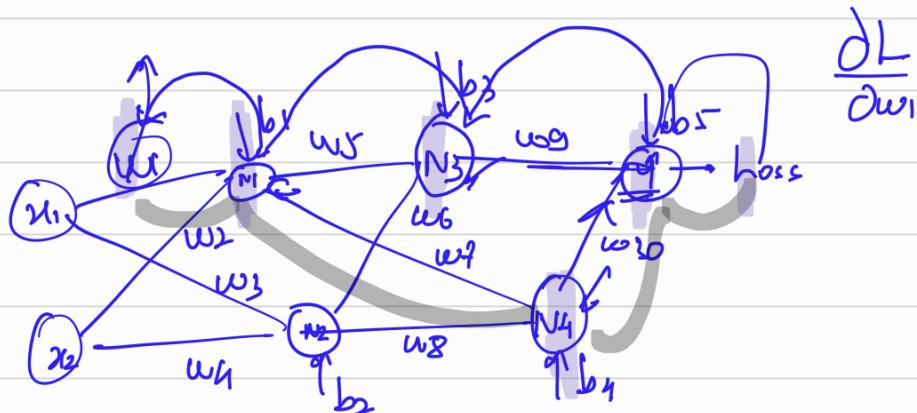
$$\frac{\partial \hat{y}}{\partial w_7} = \frac{\partial (\textcircled{w_7} \textcircled{N_1})}{\partial w_7} = N_1$$

$$\frac{\partial (x_1 x_2)}{\partial x} = 2$$



Activation functions

$$w_1x_1 + w_2x_2 + b_1$$

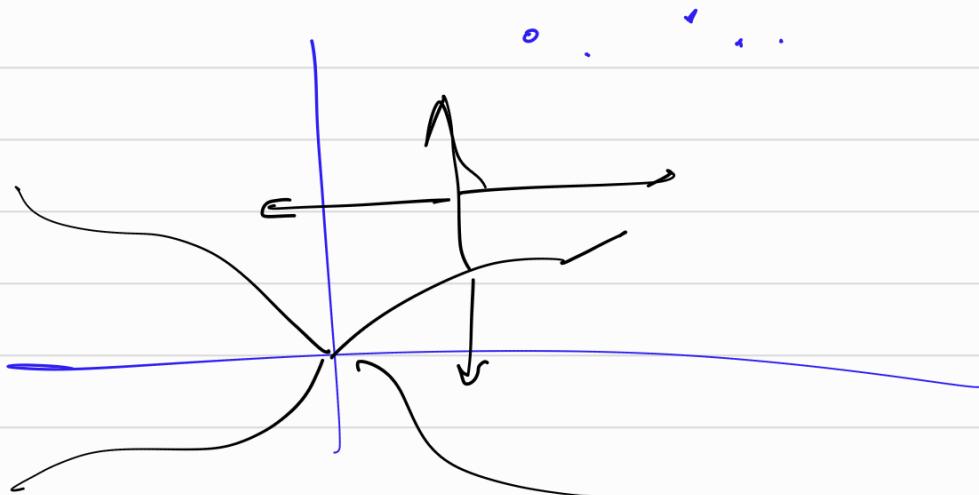


$\frac{\partial h}{\partial w_1} = \left(\frac{\partial L}{\partial y} \times \frac{\partial y}{\partial N_3} \times \frac{\partial N_3}{\partial N_1} \times \frac{\partial N_1}{\partial w_1} \right) + \left(\frac{\partial h}{\partial y} \times \frac{\partial y}{\partial N_4} \times \frac{\partial N_4}{\partial N_1} \times \frac{\partial N_1}{\partial w_1} \right)$.

$\frac{\partial L}{\partial w} = 1000000$
~~Terms = $N(N_L)^2$~~

$\frac{\partial h}{\partial w_1} = 0.00000^2$

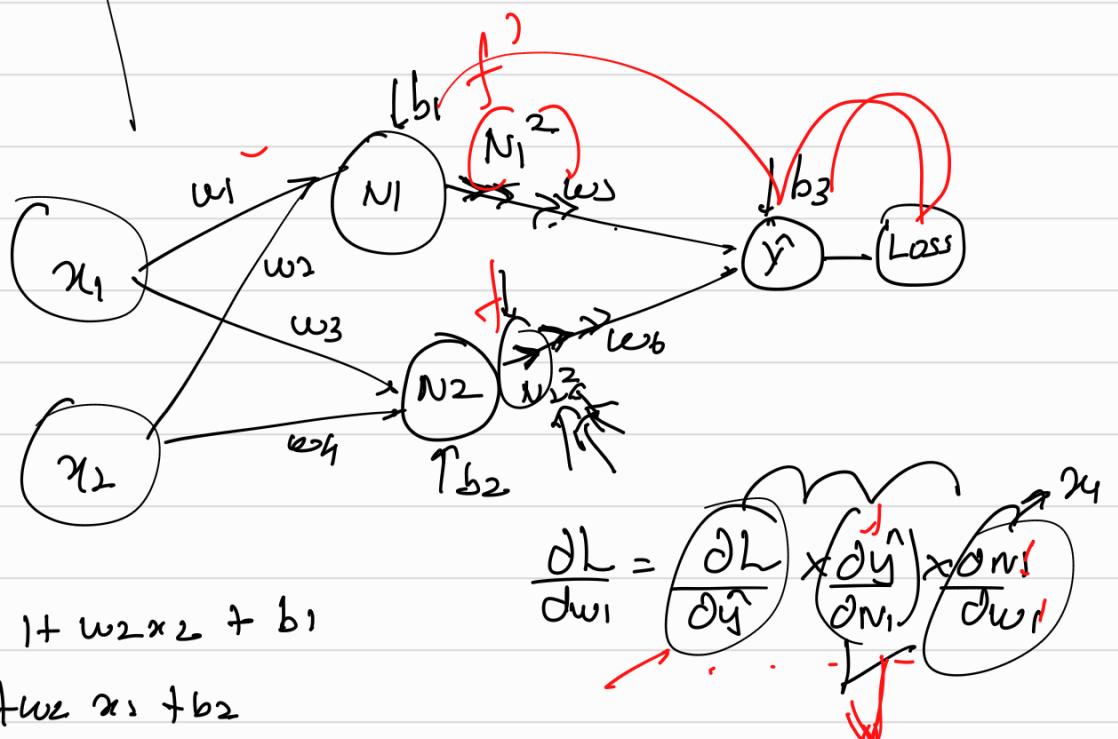
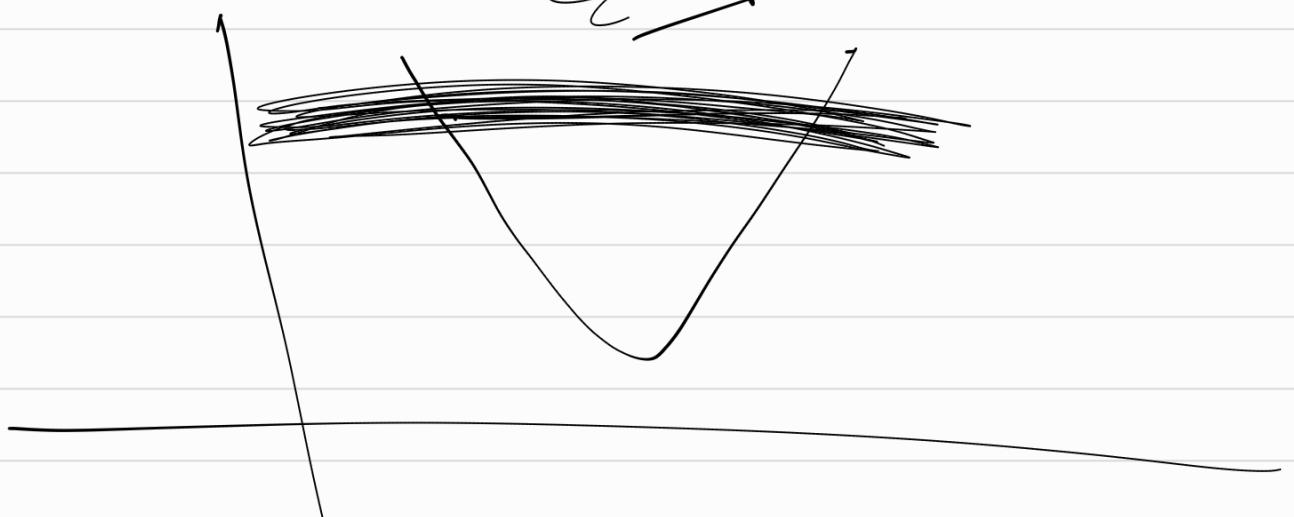
$w_1 -$



$$\left(\frac{0.99}{1.01}\right)^{365} \approx \underline{\downarrow \downarrow \downarrow}$$

$$\left(\frac{1.01}{1.00}\right)^{365} = 1.01\underline{\uparrow 37}$$

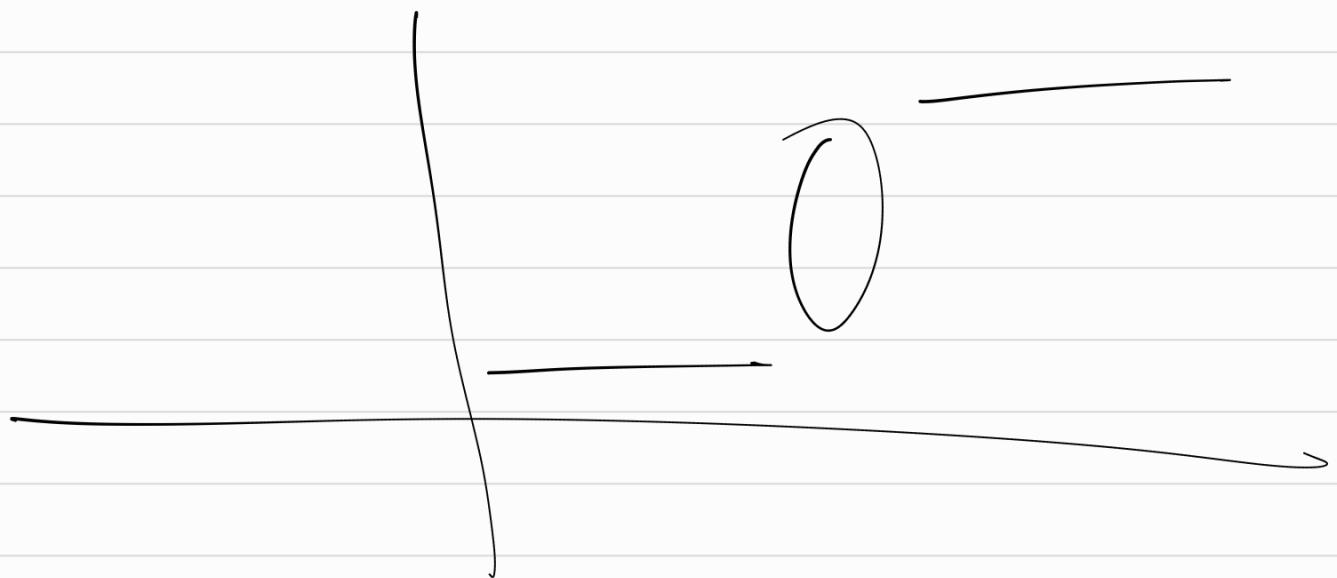
Exploding gradient



$$\hat{y} = w_5 \underline{N_1^3} + w_6 \underline{N_2^2} + b_3$$

$$\frac{\partial \hat{y}}{\partial w_1} = w_5 \times \underline{2 \times N_1}$$

~~(Activation function)~~ → Should be cont & diff.



Activation functions are functions which should be continuous & diff that can be applied at the output of any hidden layers to create non-linear feature or at the 0th to create bounded outputs → Probs

↓ output layer

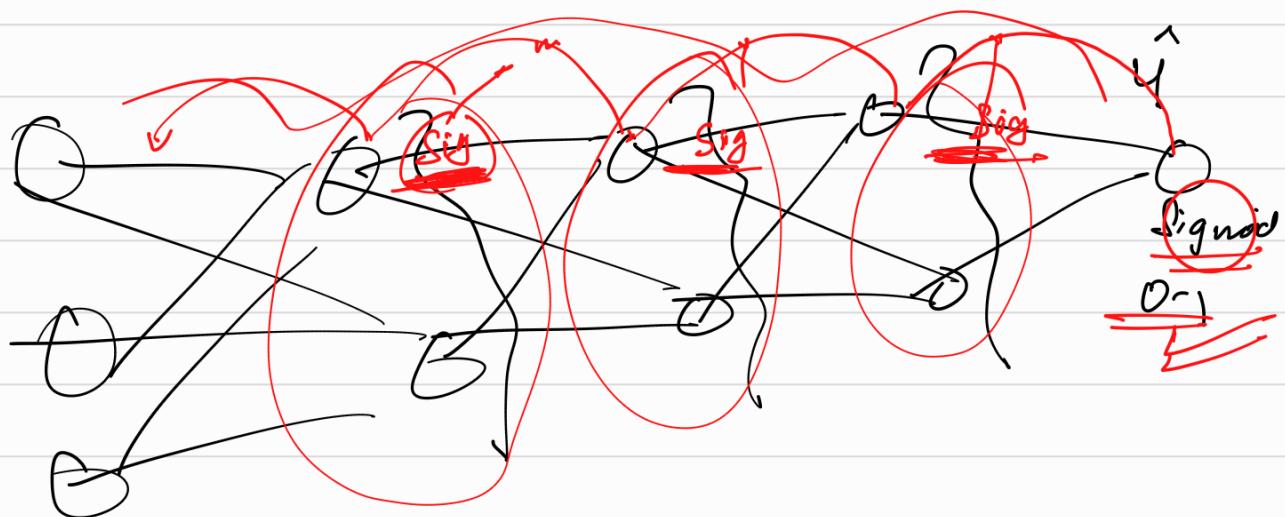
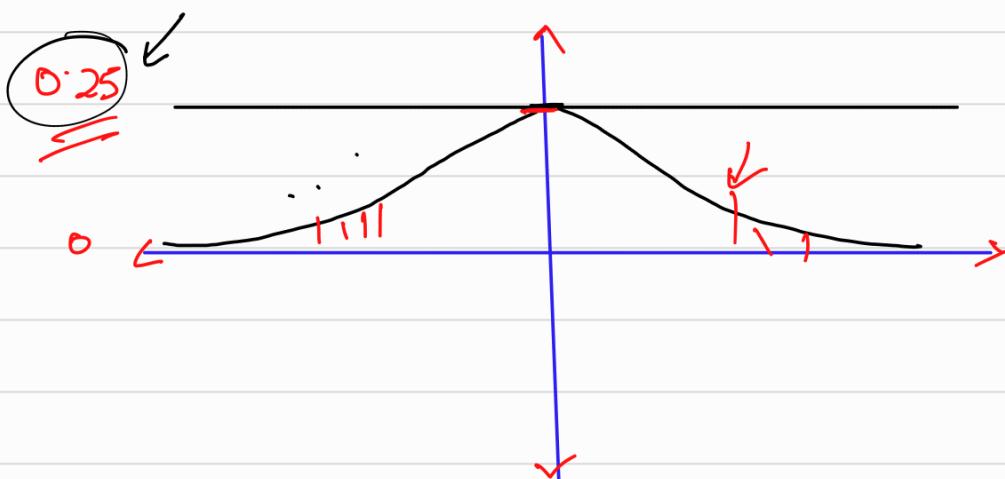
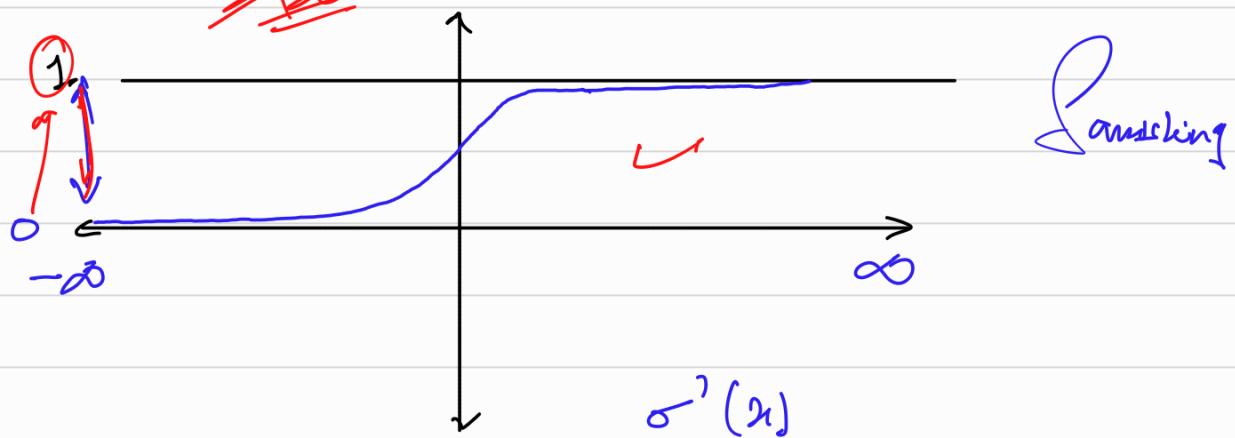
→ Sigmoid

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

$-\infty < x < \infty$

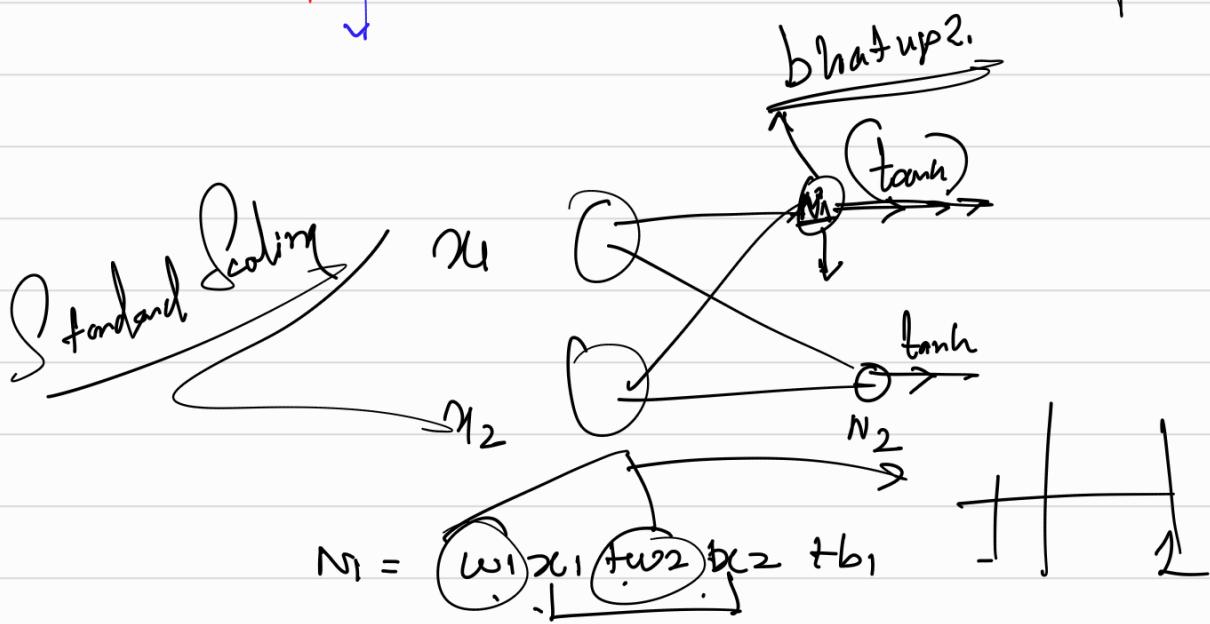
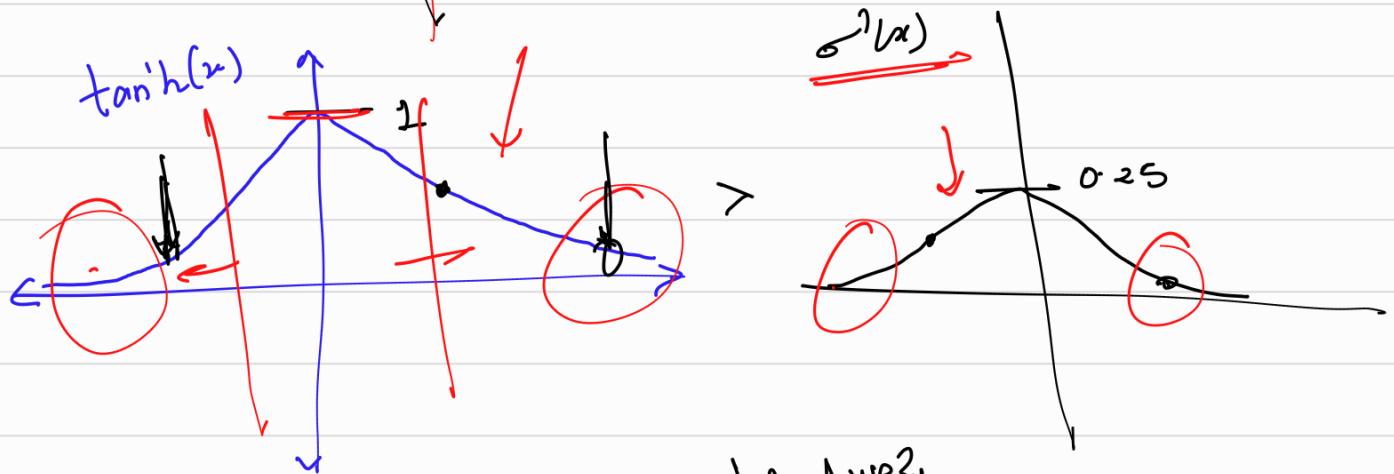
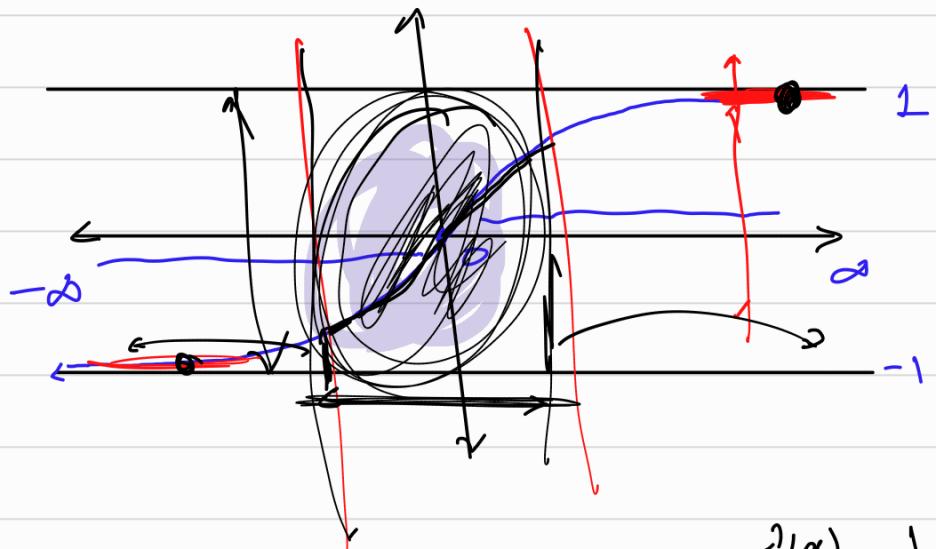
HLX

$0 < \sigma(x) < 1$

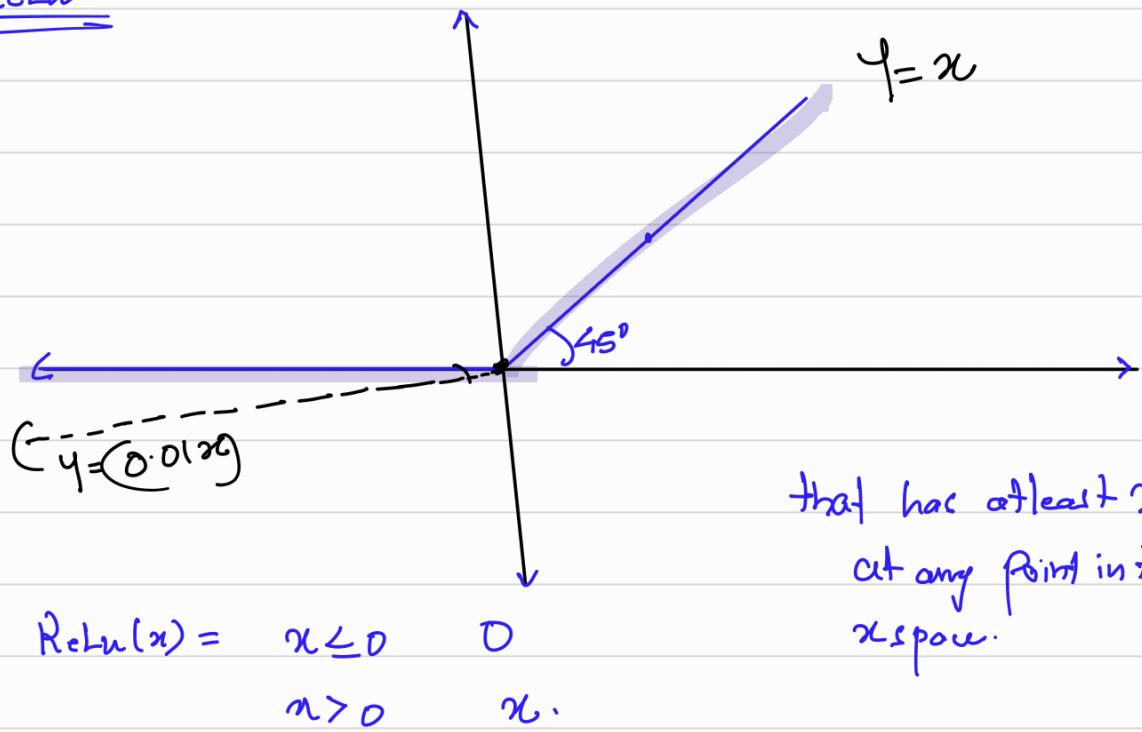


$\tanh \rightarrow$ hyperbolic tangent.

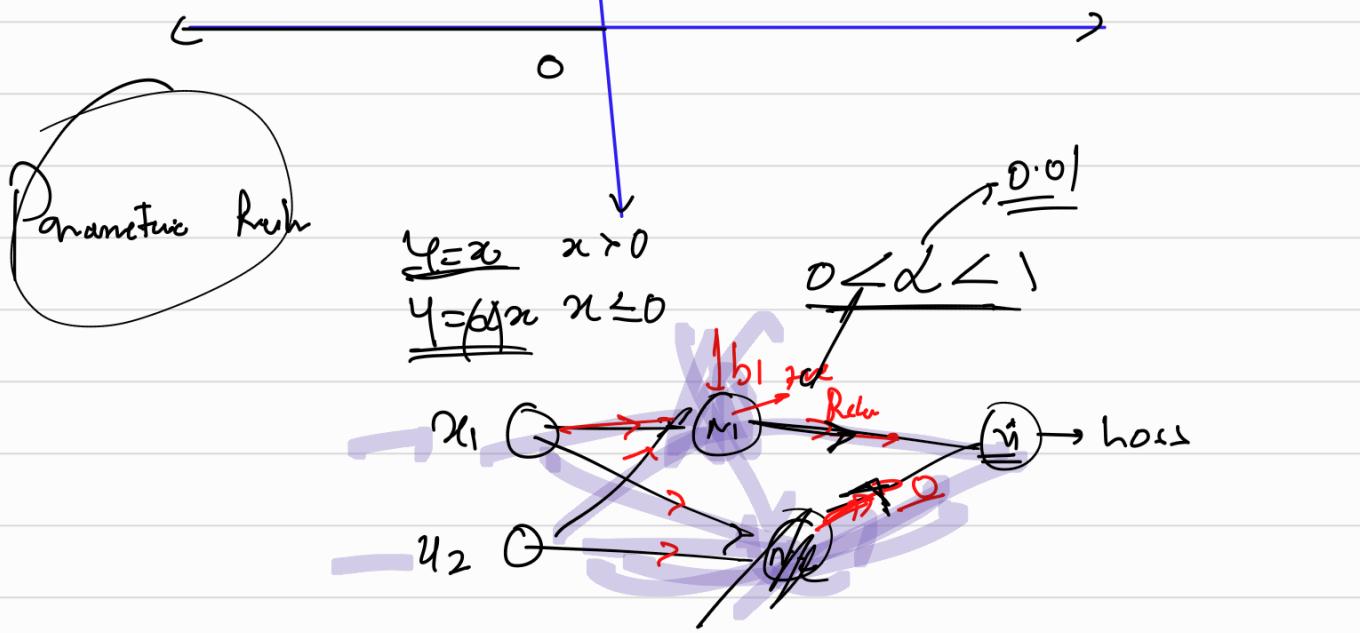
$$\tanh = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



ReLU



↳
 sketch
 $\text{ReLU}(x) \Rightarrow x \leq 0 \oplus \underline{0.01x}$
 $x > 0 \oplus 1x$



Deep Neural network

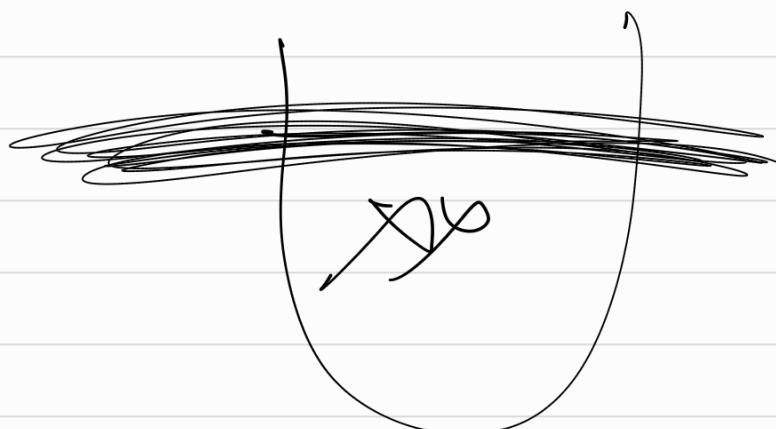
$$(2)^{\cancel{0} + \cancel{10}} = 32$$

a a a a

 2 2 2 2

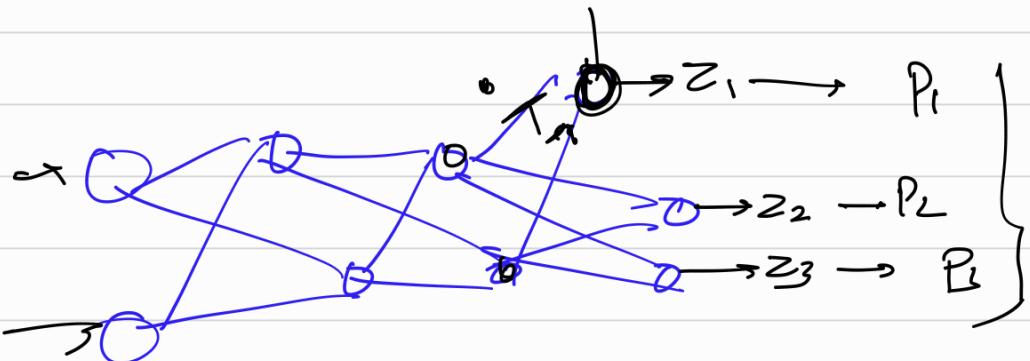
$$(2^3) \quad \underline{(2)^{100}}$$

$$\frac{dh}{d\omega_1} = w_r - \alpha \left(\frac{1}{\omega_1} \right)$$



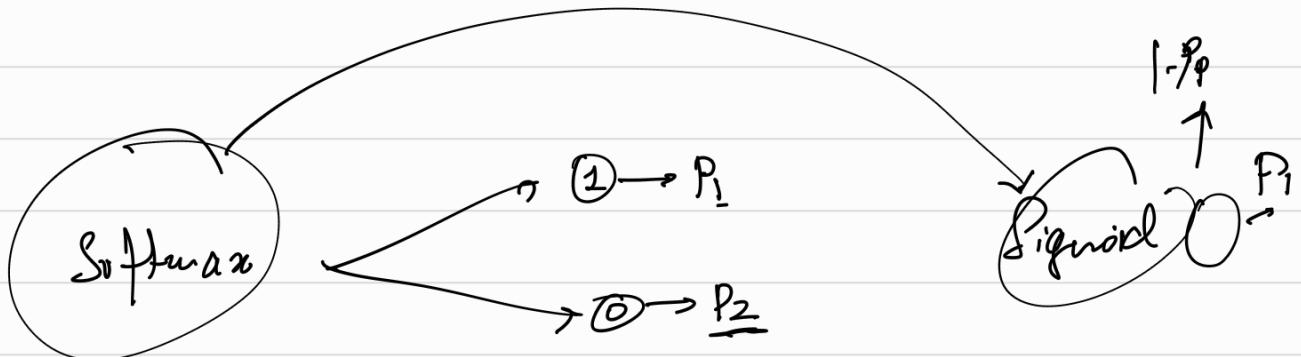
Softmax

: needs to be applied at the o/u of a multiclass problem



$$P_1 + P_2 + P_3 = 1$$

$$P_1 = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}} \quad P_2 = \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}} \quad P_3 = \frac{e^{z_3}}{e^{z_1} + e^{z_2} + e^{z_3}}$$



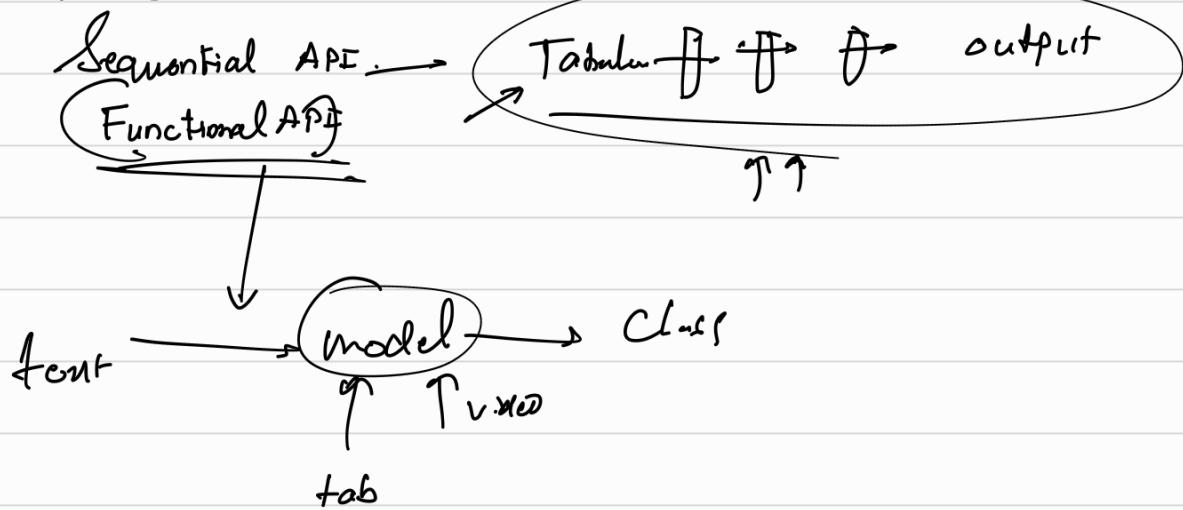
$$P_1 + P_2 = 1$$

$$P_1 = 1 - P_2$$

$$P_2 = 1 - P_1$$

Bocak

Lecture - 5:



Specce.
 $y \rightarrow LE(0, 1, 2)$
 $(y=1)$

$$\hat{y} = [0.1 \quad 0.3 \quad 0.6]$$

- $\log(\hat{y}[y])$

$y \rightarrow DCE.$
 $y \rightarrow [0, 1, 0].$

$cce.$

$$\hat{y} = [0.1 \quad 0.3 \quad 0.6]$$

$$y = [0 \quad 1 \quad 0]$$

$$- [0 \times \log 0.1 + 1 \times \log 0.3 + 0 \times \log 0.6]$$

$$= - \log 0.3$$

$$\frac{\partial f}{\partial x_i}$$

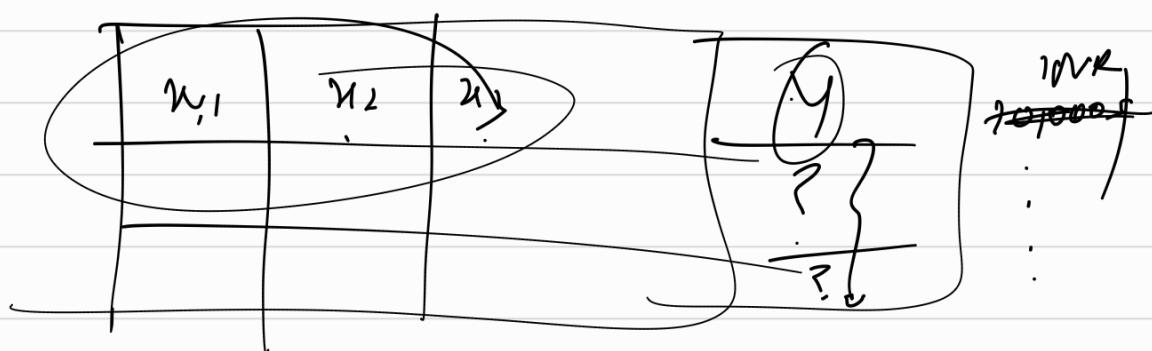
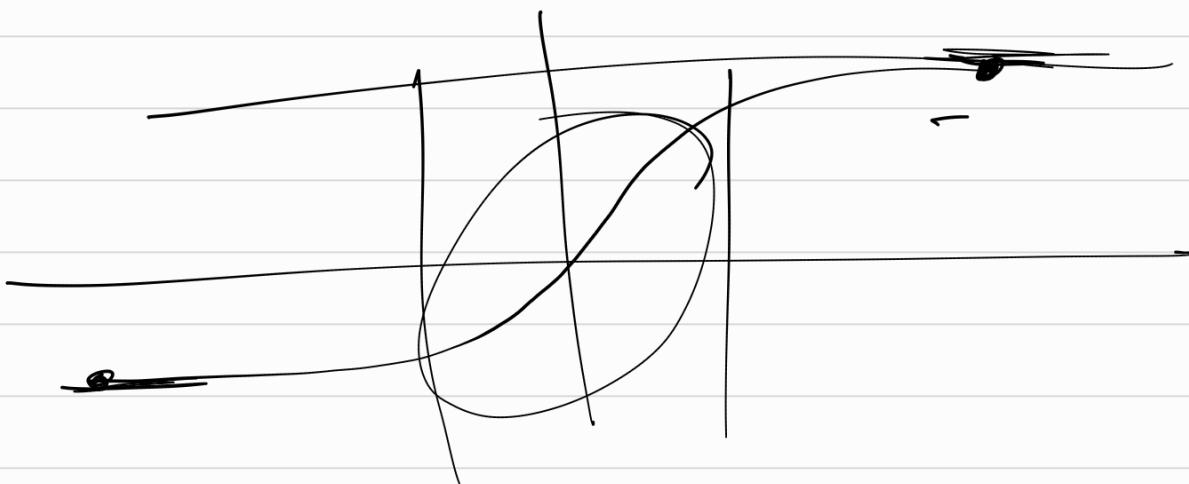
$$0$$

Owner	Kms	Mileage	y
w_1	w_2	w_3	
$w_1 u$	$w_2 v$	$w_3 w$	
w_1	w_2	w_3	

$$w_1 \times \underline{\text{Owner}} + w_2 \times \underline{\text{Kms}} + w_3 \times \underline{\text{Mileage}}$$

(1) Honda City
50,000 ↘

(2) Honda City
40,000



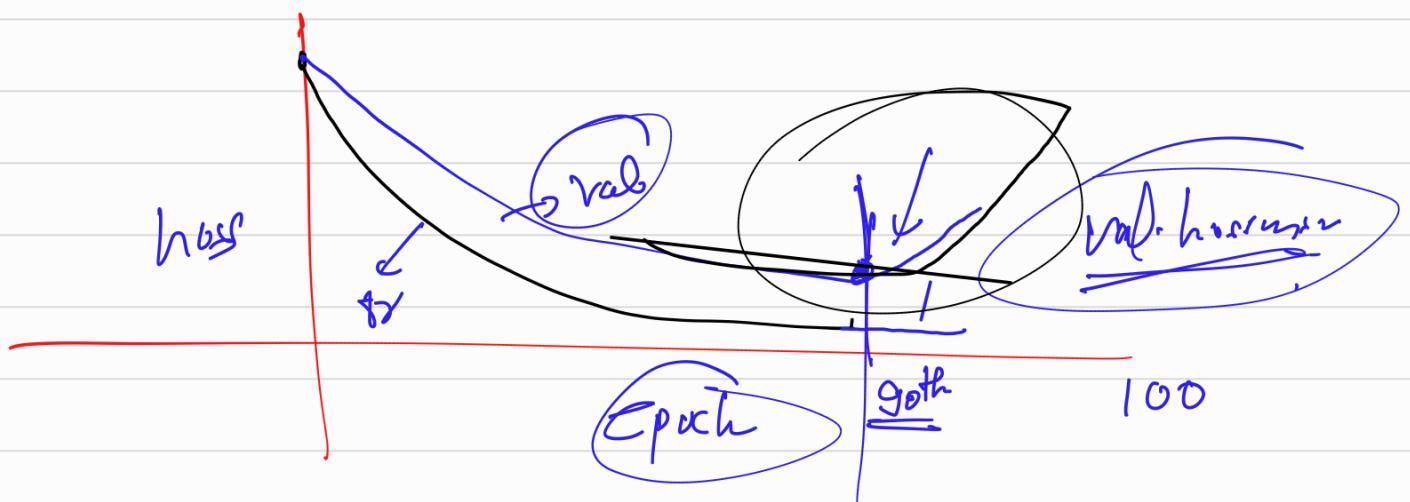
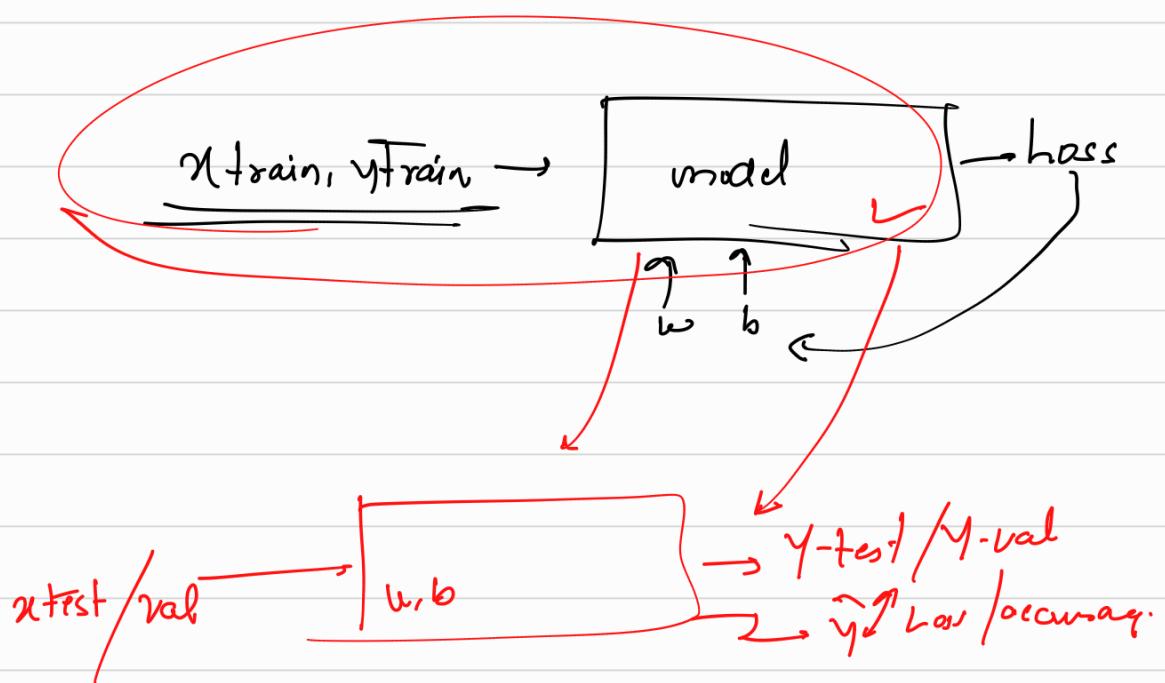
3,-3
($w_2 \sim -1, 1$)

$\begin{matrix} 3,-3 \\ 3,1 \\ -1 \end{matrix}$

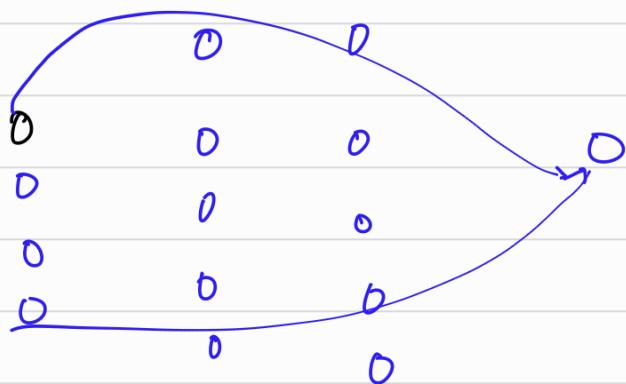
$2+1 \rightarrow$

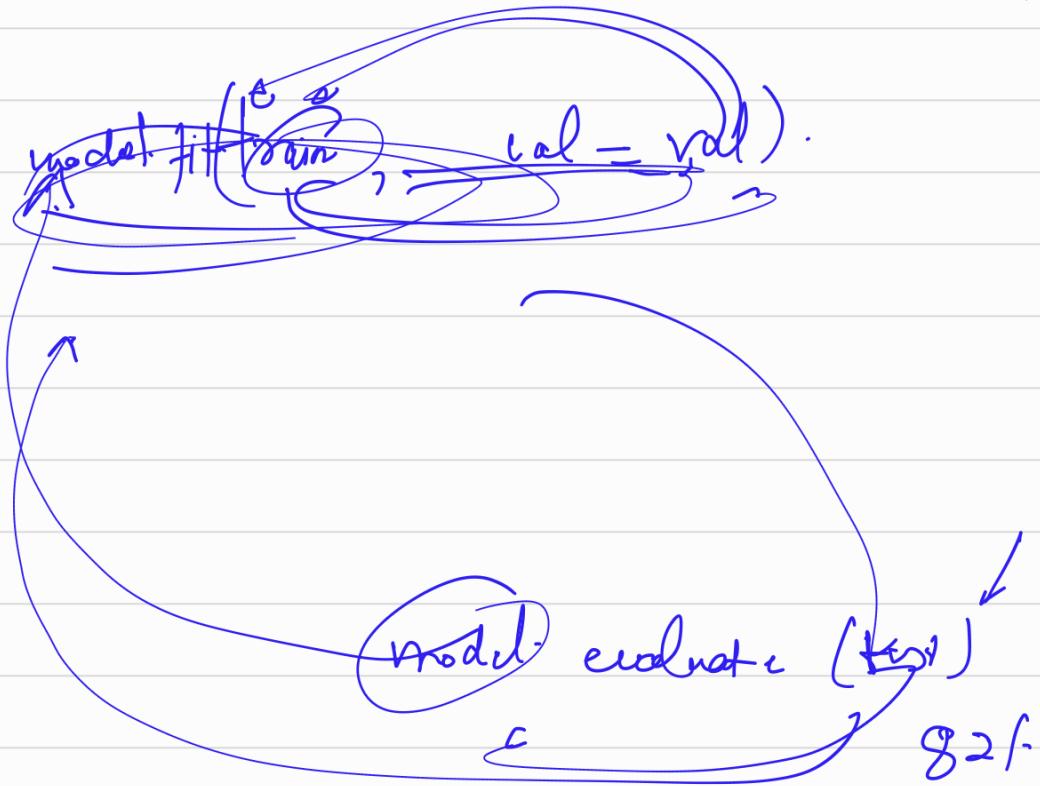
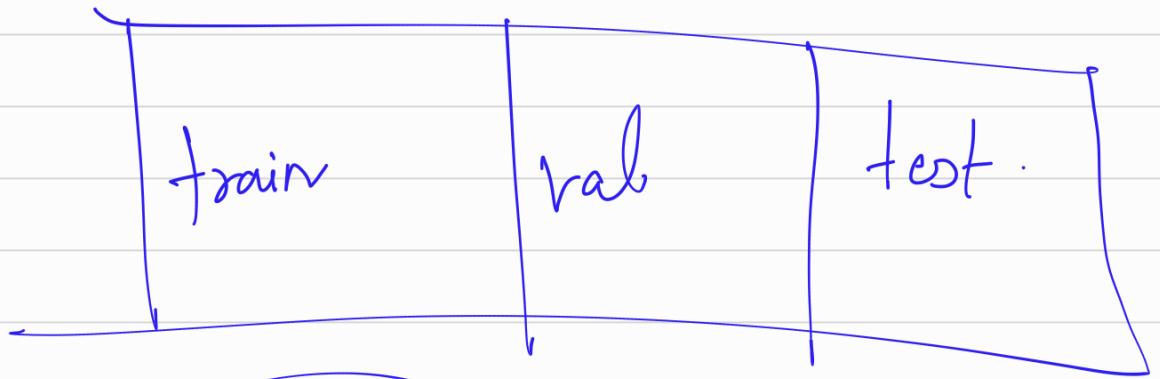
$0 \rightarrow \underline{\underline{50,000}}$

$L = (\underline{\underline{50,000 - 0.9}})^2$



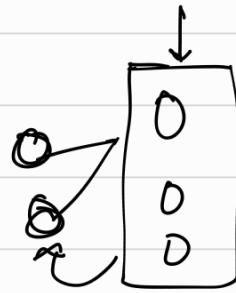
Callbacks



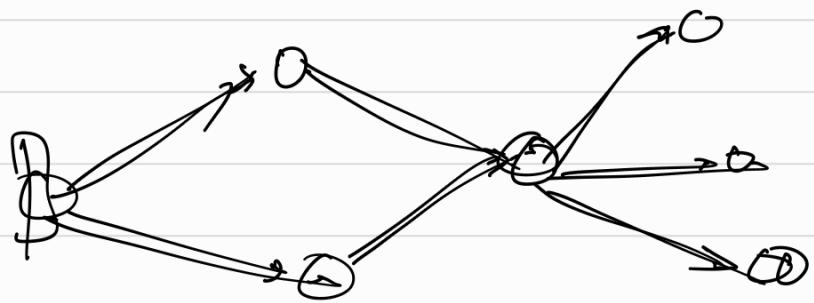


Functional API:

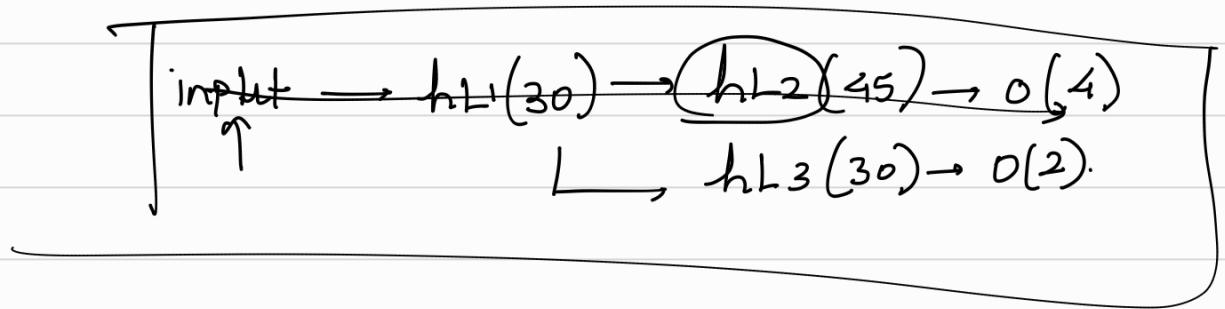
Graph Structure



any layer (ϵ)



$D \rightarrow D$



Input $\xrightarrow{(4)} hL_1(30) \rightarrow hL_2(40)$

Case 1: Input $\xrightarrow{(4)} hL_1(20) \rightarrow hL_2(2) \rightarrow 50$

Case 2:

inp1 = Input(4,)

inp2 = Input(4,)

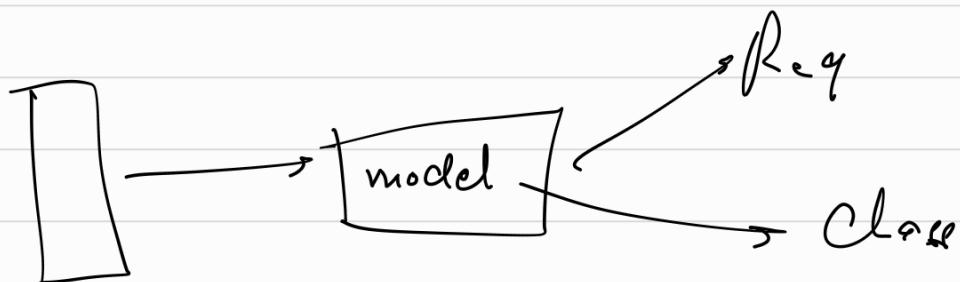
hL1 = Dense(30)(inp1)

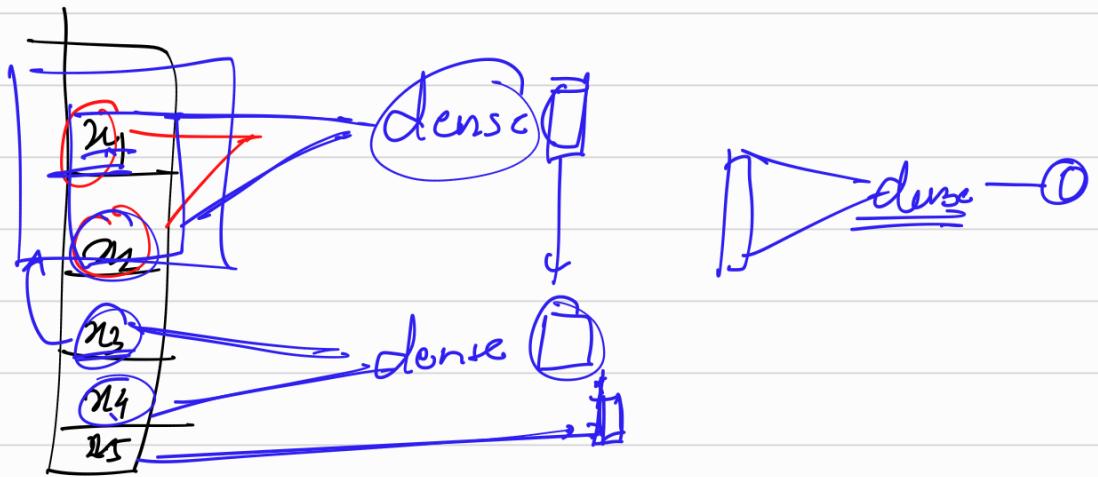
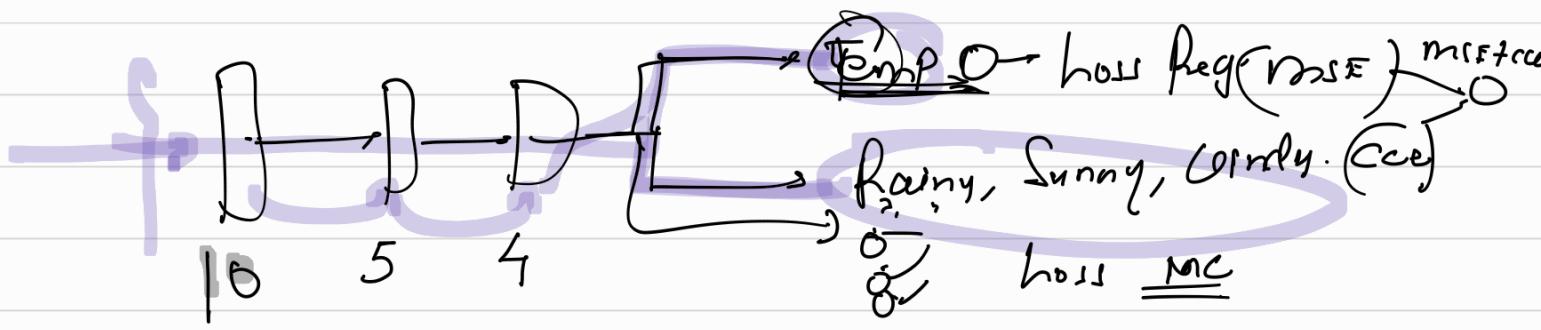
hL2 = Dense(40)(hL1)

hL3 = Dense(20)(inp2)

hL4 = Dense(50)(hL3)

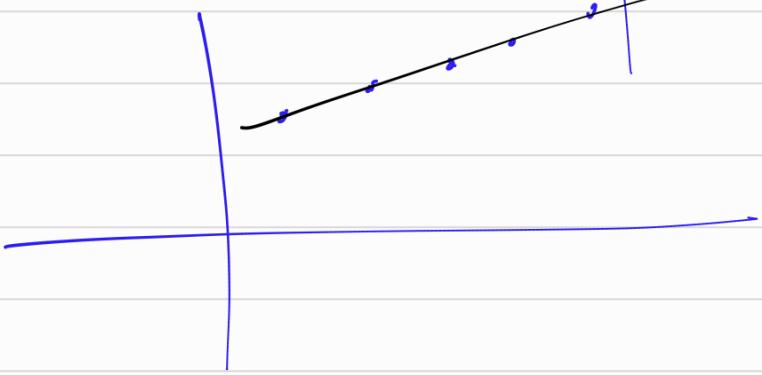
O = Concatenate([hL2, hL4])

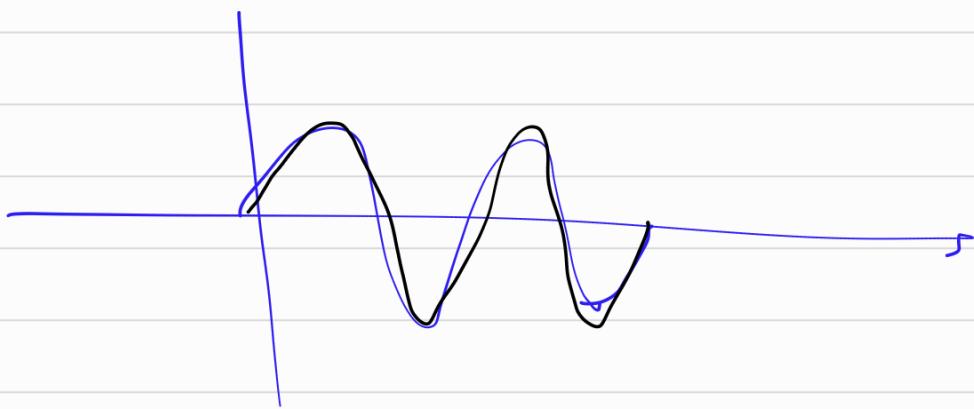
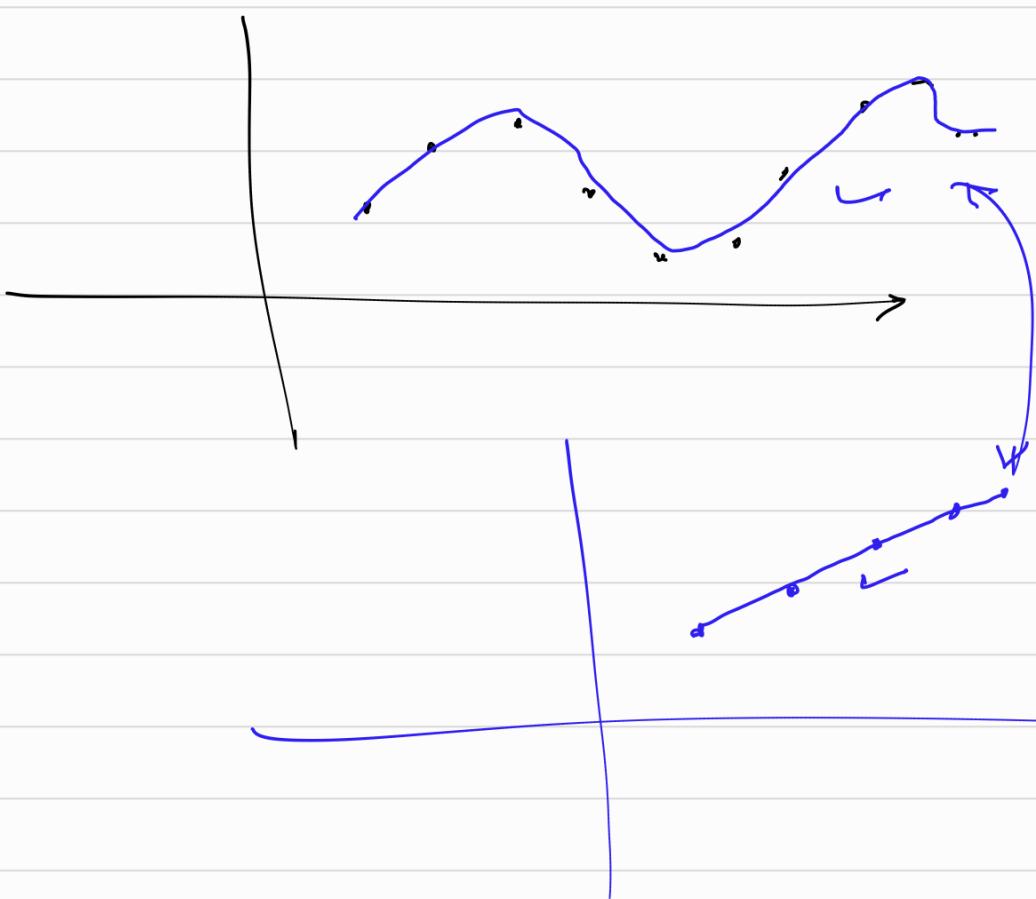
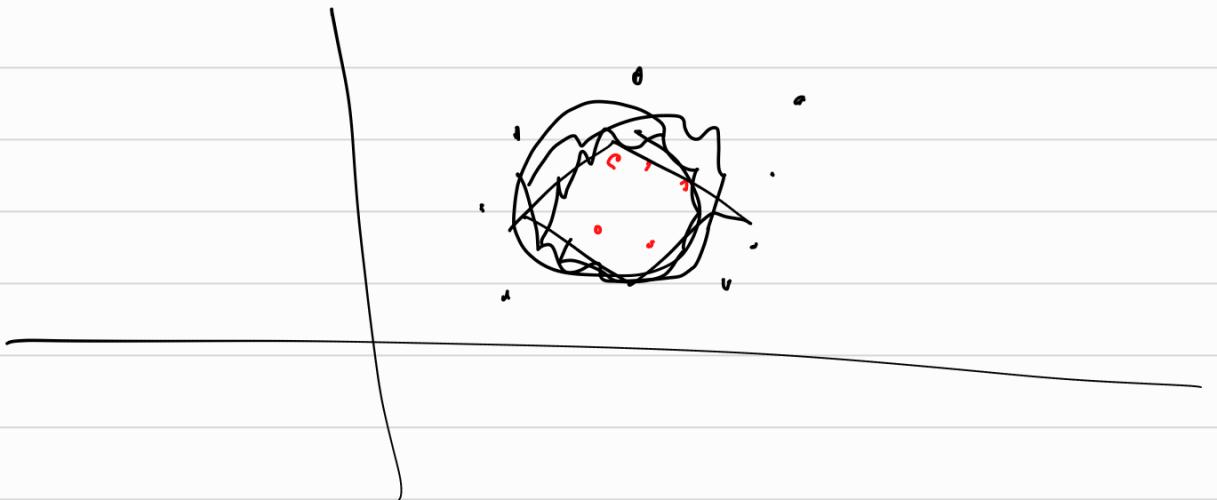




Optimizers

loss





gg. gggg).



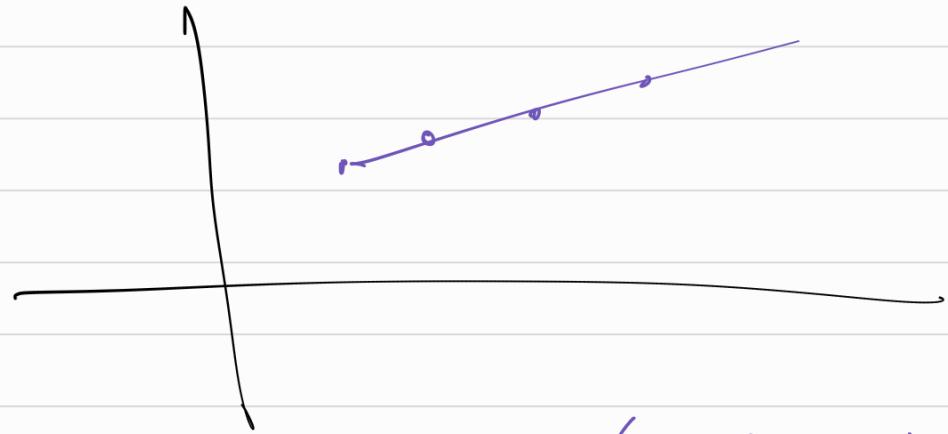
(Regularization.)

- Regularization
- Dropout
- Optimizers.
- Callbacks
- Batch Normalization
- weight initialization

Hyperparameter Tuning

Regularization:

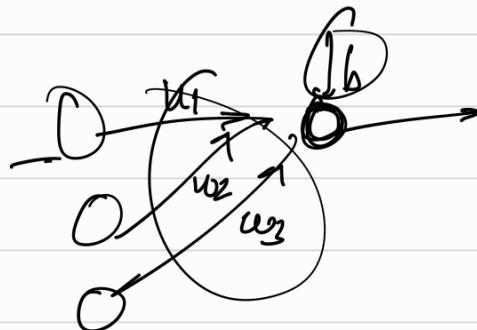
↓ ↓
(Ridge and Lasso.) ?



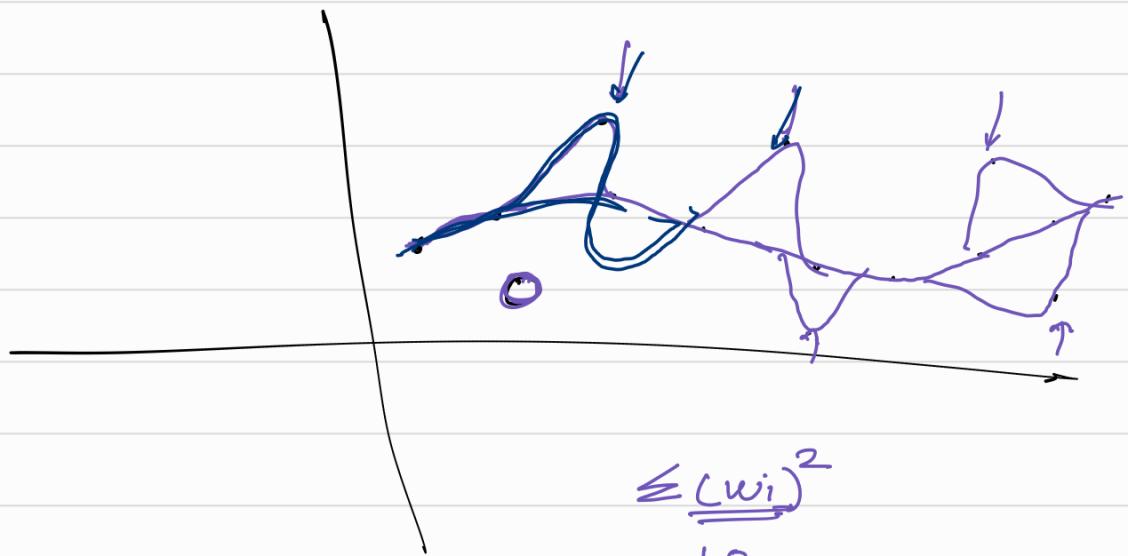
Overfit → has high variance (model is extremely sensitive to changes in n data).

$$\textcircled{1} \quad y = \underline{100x_1} + \underline{500x_2} \quad y=1000$$

$$\textcircled{2} \quad y = 2x_1 + x_2 = \textcircled{2}$$



$$N1 = \underline{w_1} \underline{x_1} + \underline{w_2} \underline{x_2} + \underline{w_3} \underline{x_3} + b_1$$



TL mSE

$$510 = 500$$

$$425 = 400$$

$$375 = 320$$

$$355 = 260$$

$$320 = 200$$

$$305 = 150$$

$$\underline{\underline{345 = 145}}$$

$$\leq \frac{(w_i)^2}{10}$$

$$25$$

$$55$$

$$95$$

$$= 120$$

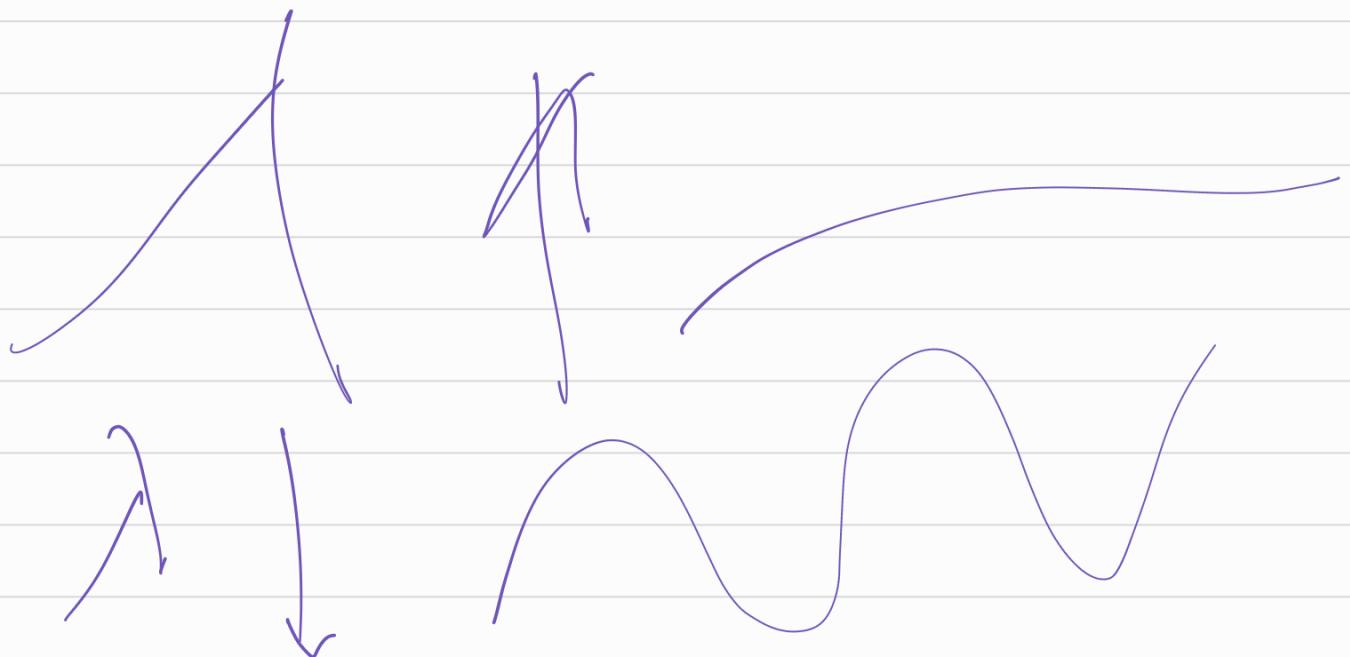
$$= \frac{155}{200} \}$$

$$TL = \text{mse} + \boxed{\frac{1}{g}} \sum_{j=1}^p w_j^2$$

Lagrangean multiplier

↓
Hyperparameter (0 - ∞)

↓
~~Val-loss~~



$\lambda =$

$\lambda = 0.5$

$\lambda = 0.9$

$\lambda = 0.95$

$\lambda = 0.5$

$\rightarrow 92\%$

$\rightarrow 95\%$

Reg.

$$T_L = \text{mse} + \lambda \sum w_j^2 \quad \text{Ridge } (L^2).$$

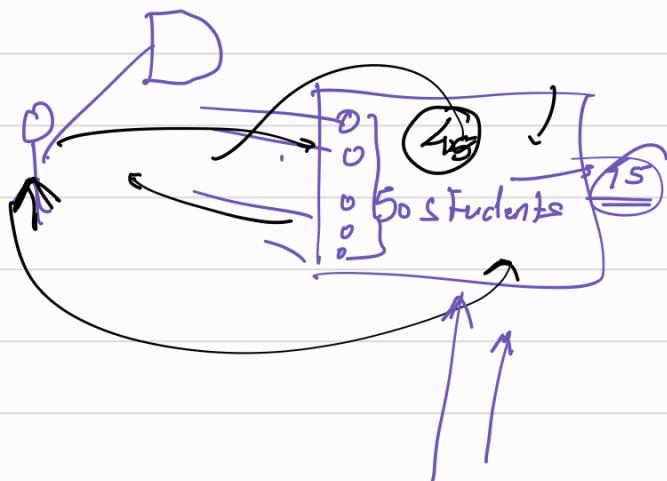
$$T_L = \text{mse} + \lambda \sum |w_j| \quad \text{Lasso } (L_1).$$

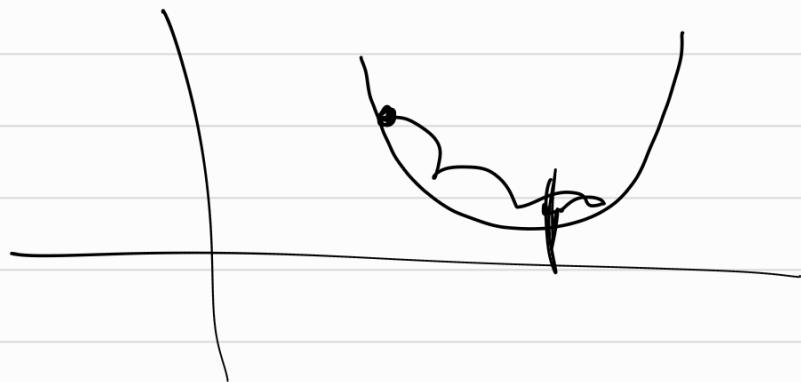
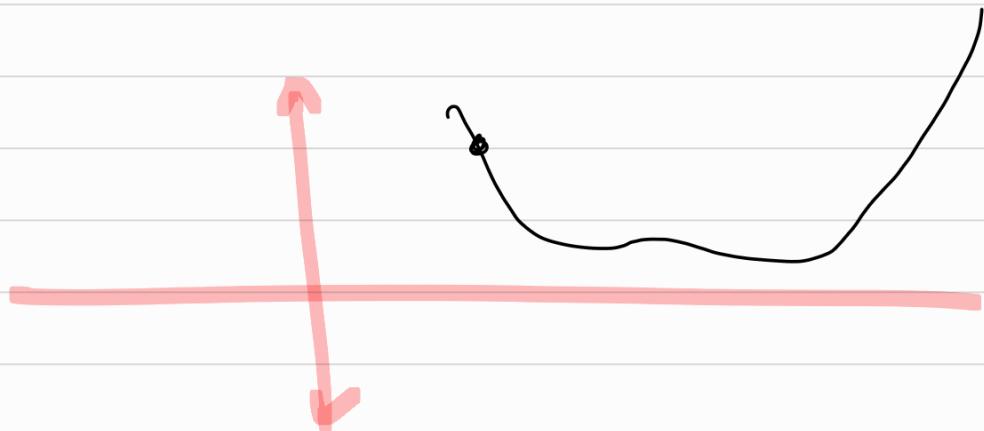
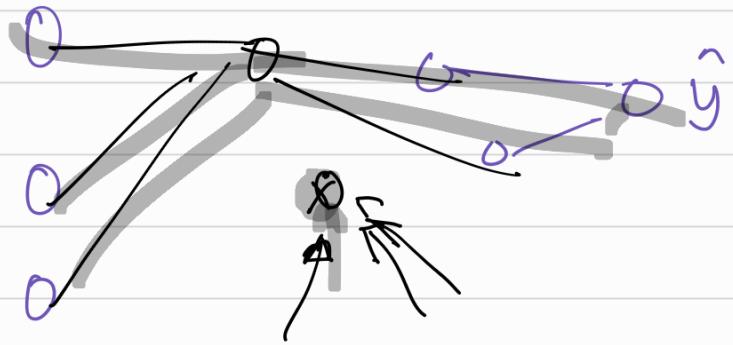
$$T_L = \text{mse} + \lambda_1 (\sum w_j^2) + \lambda_2 \sum |w_j| \quad L_1 L_2$$

Elastic Net

Break!

Dropout:





10

