Deep learning

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

ML

Reg

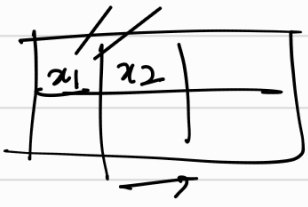
Class.

LR, LR ← Parametric models

DT, RF, Xgboost ← Non-parametric

Process

$y \sim f(x)$

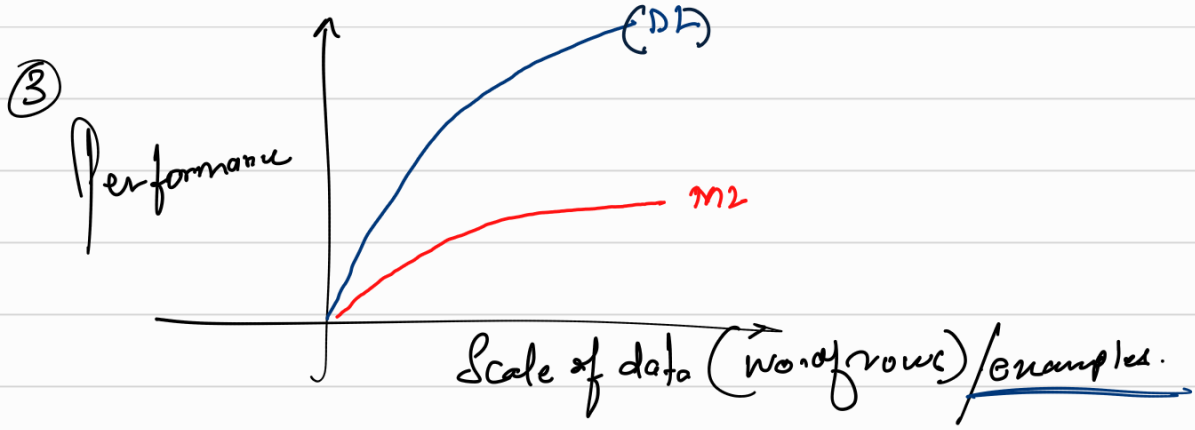


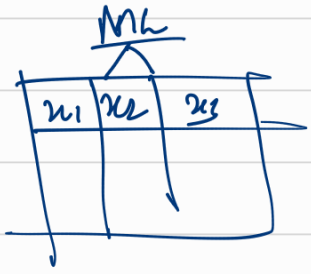
① $x \rightarrow$ (video, audio, text, tabular data, graph).

architecture

multimodality

② Tabular data, text, Scan → model → Diagnosis



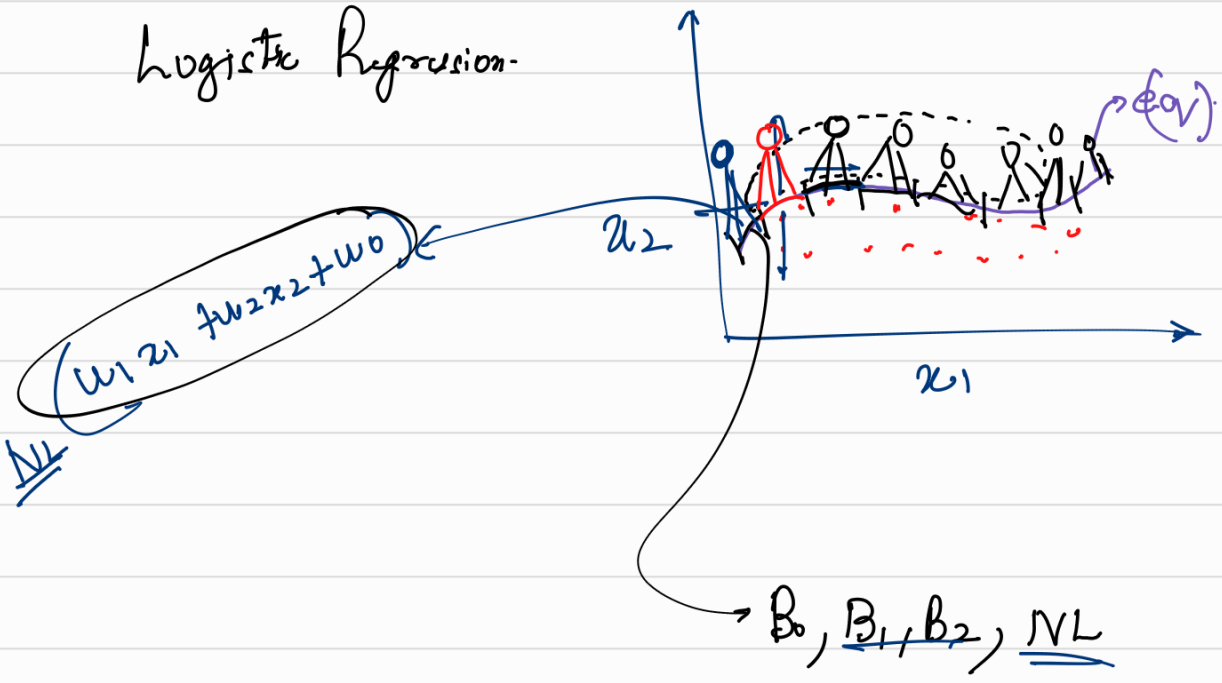


DL → automated feature engineering

DL

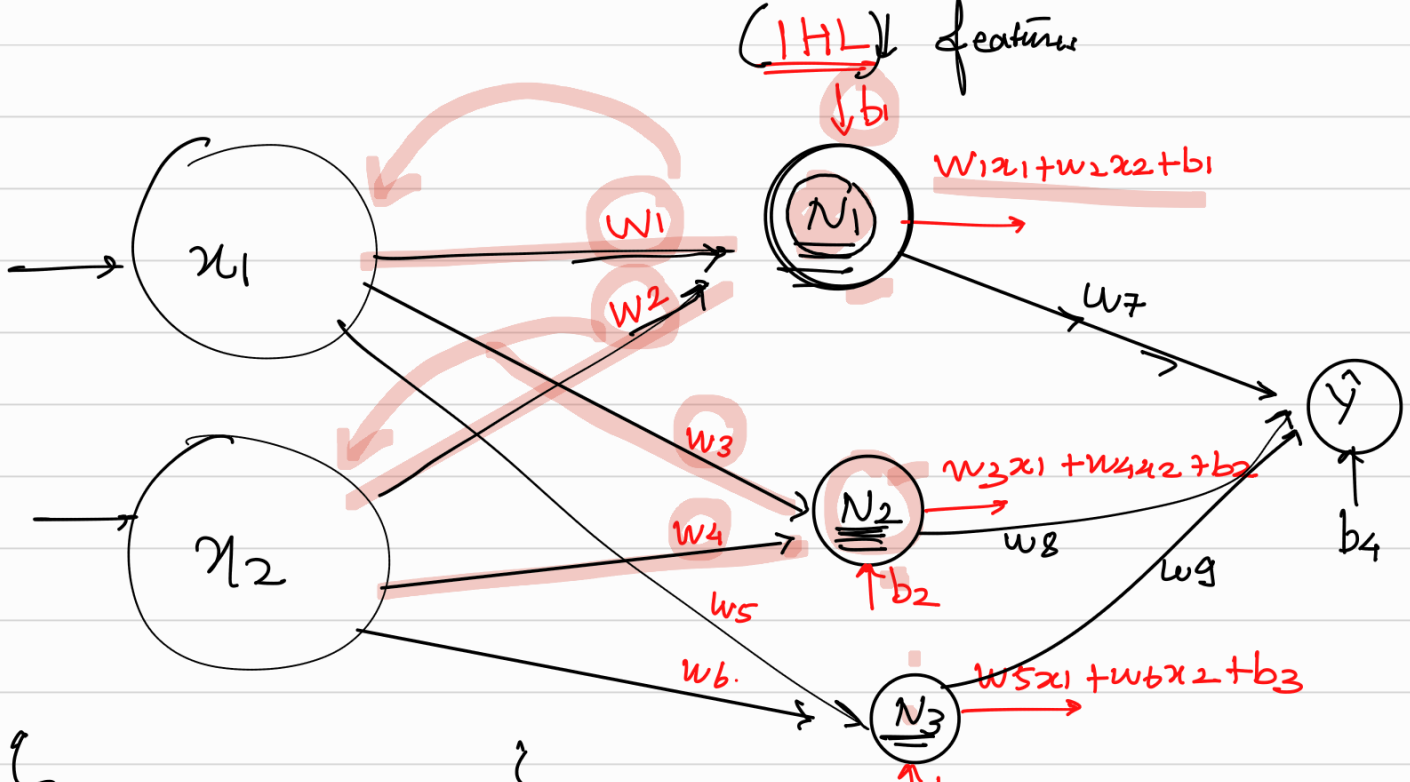
What is a Neuron

Logistic Regression.



B_0, B_1, B_2, NL

x_1	x_2	y



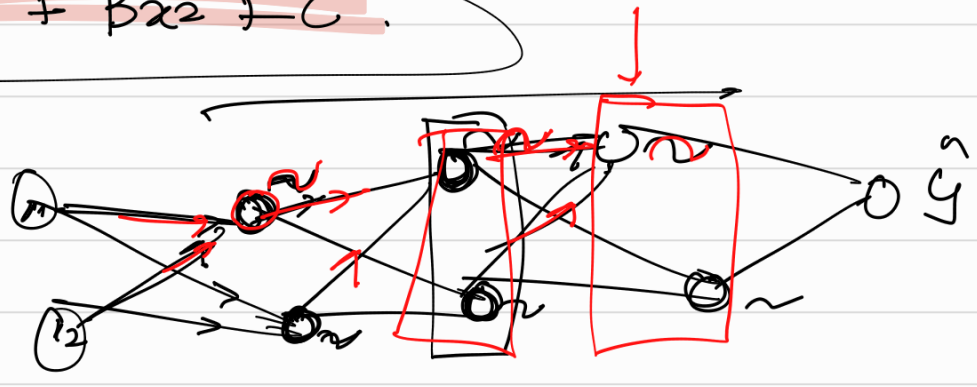
$N_1 = (w_1x_1 + w_2x_2 + b_1)$, $N_2 = (w_3x_1 + w_4x_2 + b_2)$ $N_3 = (w_5x_1 + w_6x_2 + b_3)$

$$\hat{y} = w_7N_1 + w_8N_2 + w_9N_3 + b_4$$

$$\Rightarrow \hat{y} = w_7(w_1x_1 + w_2x_2 + b_1) + w_8(w_3x_1 + w_4x_2 + b_2) + w_9(w_5x_1 + w_6x_2 + b_3) + b_4$$

$$\Rightarrow \hat{y} = x_1(w_7w_1 + w_8w_3 + w_9w_5) + x_2(w_7w_2 + w_8w_4 + w_9w_6) + w_7b_1 + w_8b_2 + w_9b_3 + b_4$$

$$\Rightarrow \hat{y} = Ax_1 + Bx_2 + C$$



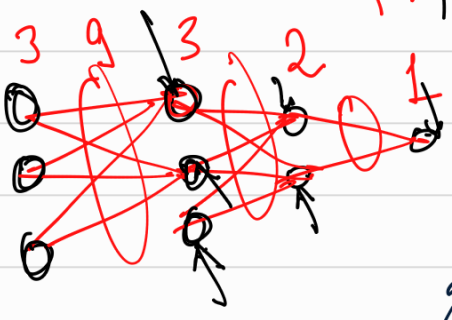

1 HL
nth HL

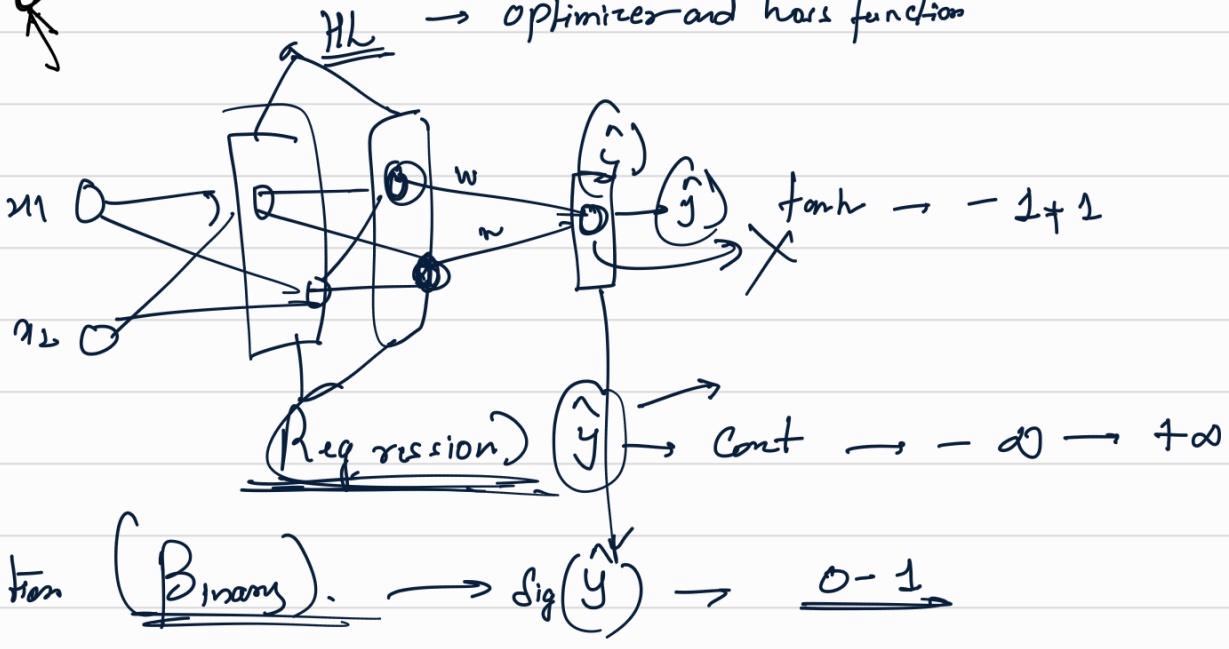
Gradient Descent : minimise the loss by updating the W, B. for all neurons.

Random

1) Define the neural network architecture

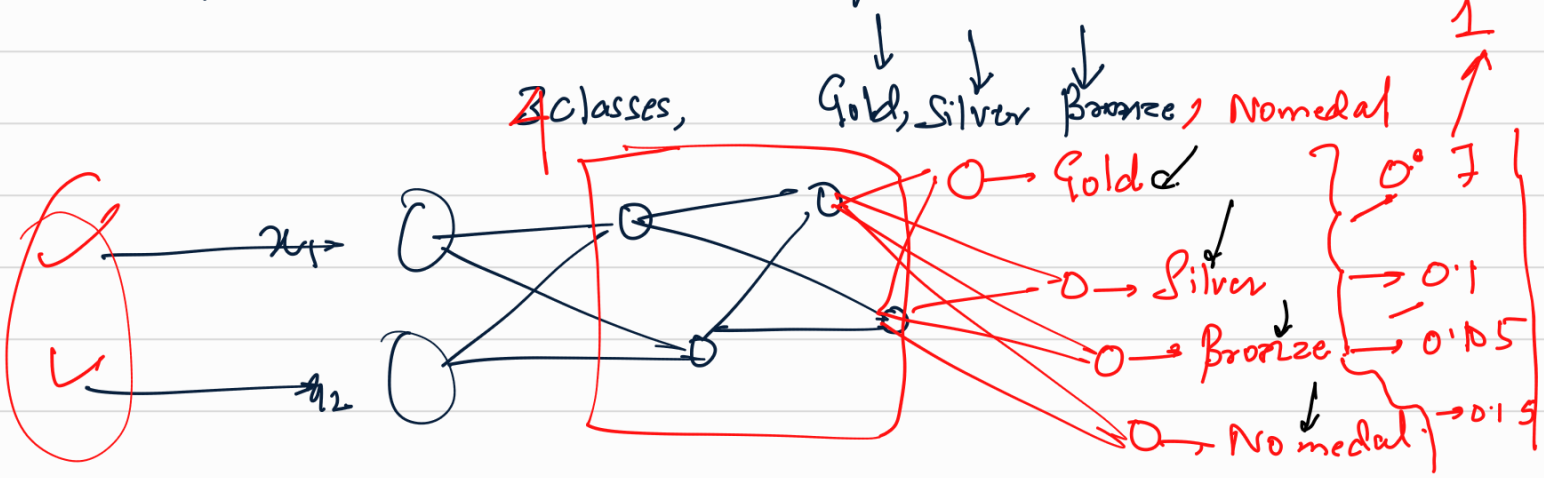
W, b → HLs → Hidden layers
→ Neurons
→ activation function
→ weights and biases → Random
→ optimizer and loss function

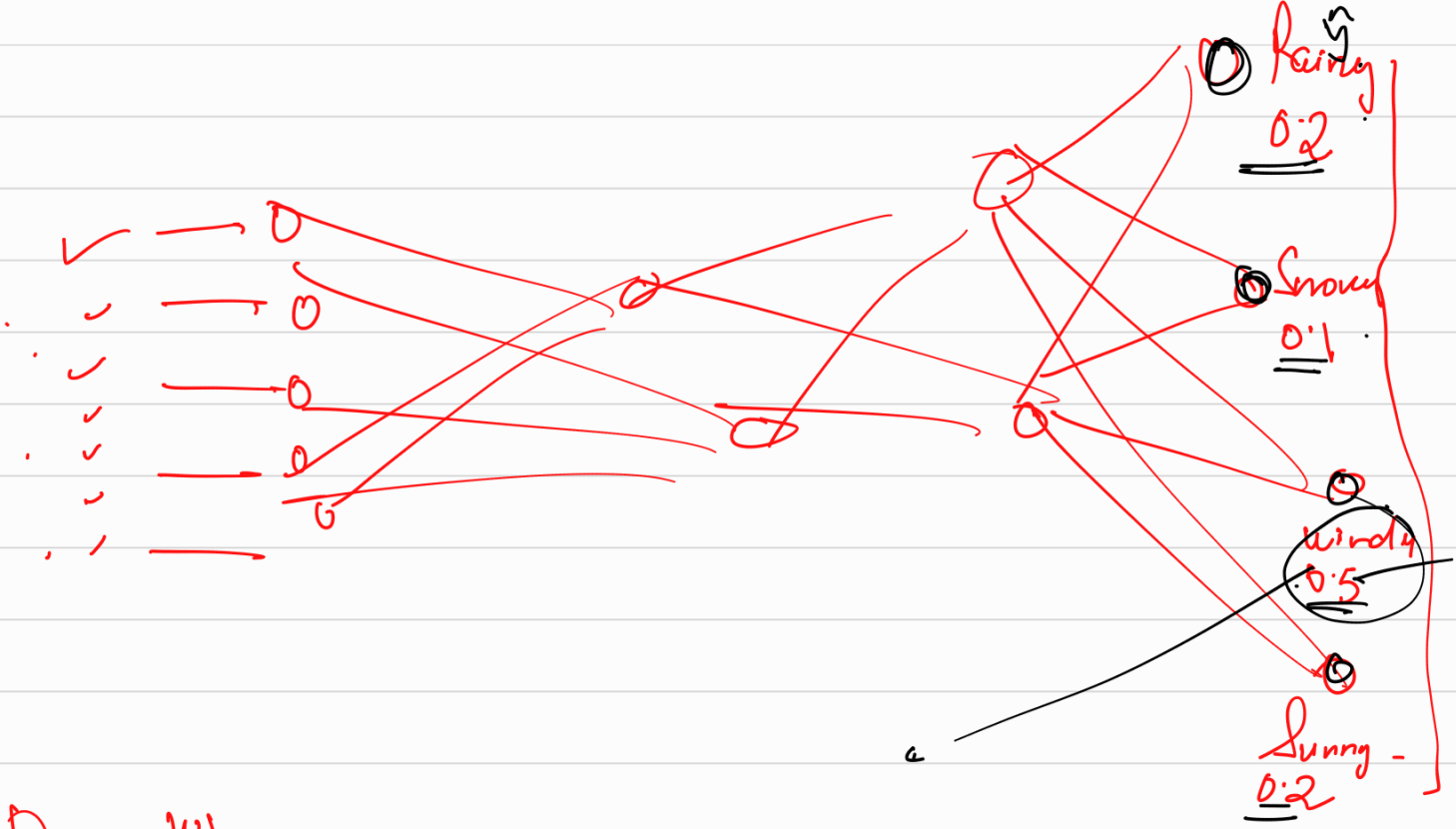




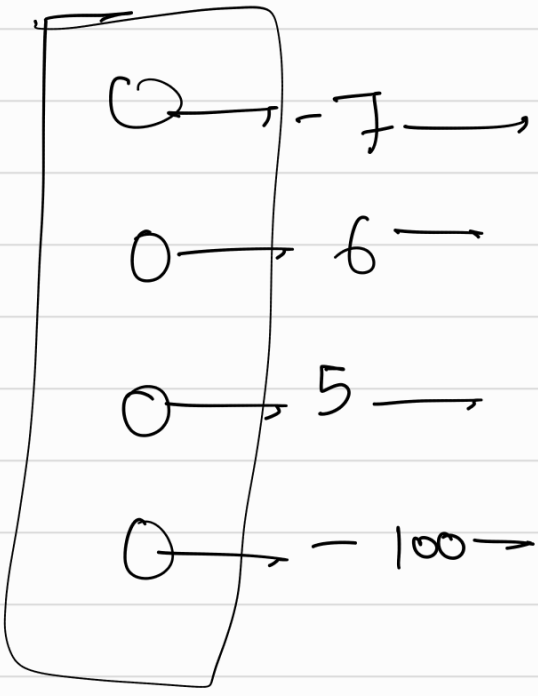
Classification (Binary). → sig(ŷ) → 0-1

Classification (multiclass). Softmax (ŷ).

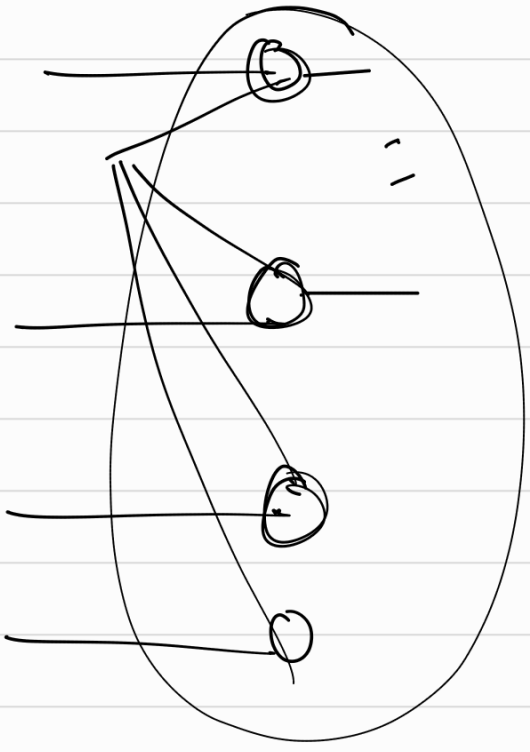


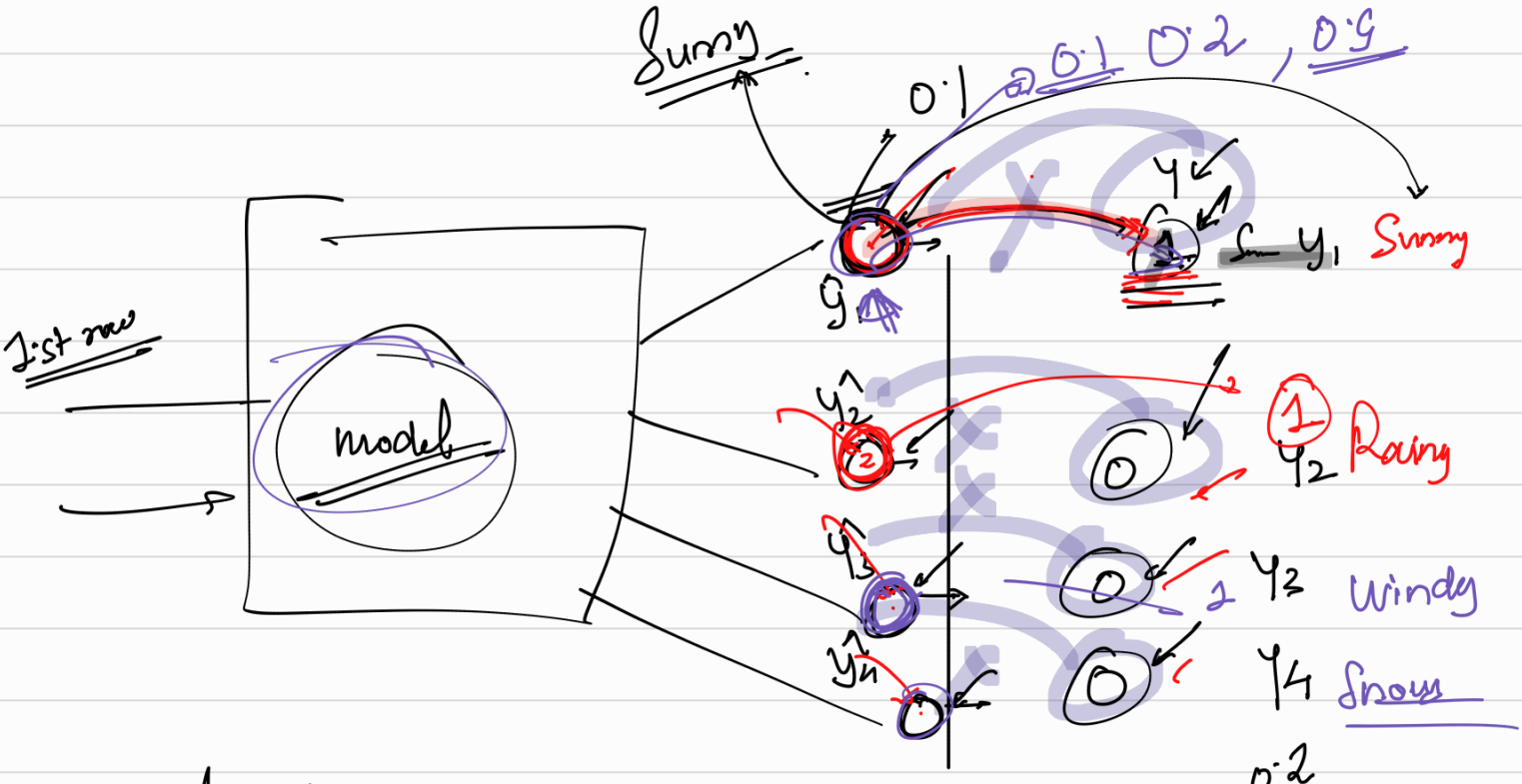

Rainy
0.2
Snowy
0.1
Windy
0.5
Sunny
0.2


x
w1
x
w


7
6
5
100

x_1	x_2	weather (y)	y_Su	y_Ra	y_Wi	y_Sn
		Sunny	1	0	0	0
		Rainy	0	1	0	0
		windy	0	0	1	0
		Sunny	1	0	0	0
		Snowy	0	0	0	1





Log loss = for all rows of data sum $-\left(y_1 \log(\hat{y}_1) + y_2 \log \hat{y}_2 + y_3 \log \hat{y}_3 + y_4 \log \hat{y}_4\right)$.

$\hat{y}_1 = 0.2$, $\hat{y}_2 = 0.8$

$- (\log 0.2)$

$= 0.69$

$- (\log 0.8)$

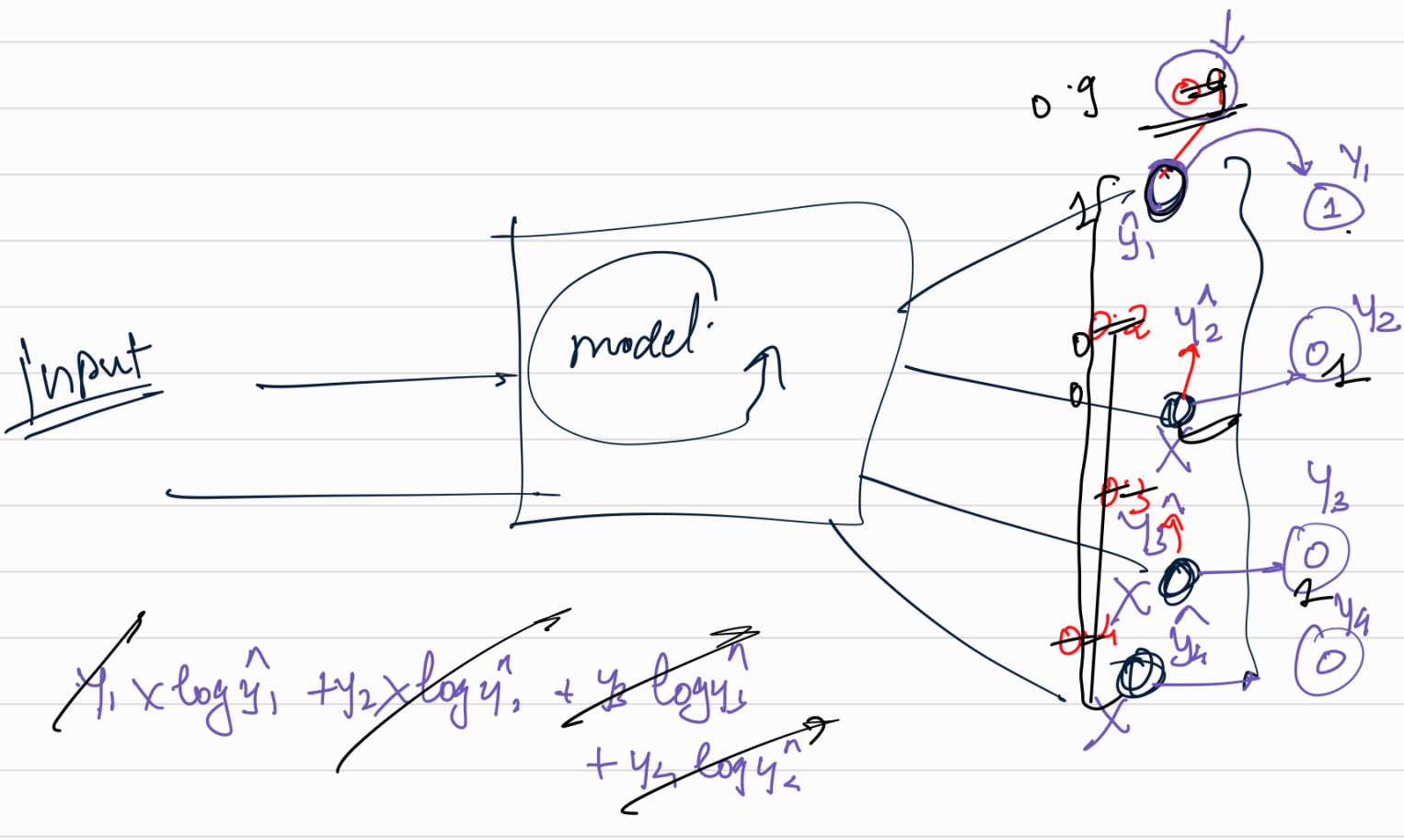
$= 0.09$

$- \log(0.95) = 0.02$

$\log 1 = 0$

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

$$\log(0.1) = \underline{1}$$



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

$\log 1 = 0$

①

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

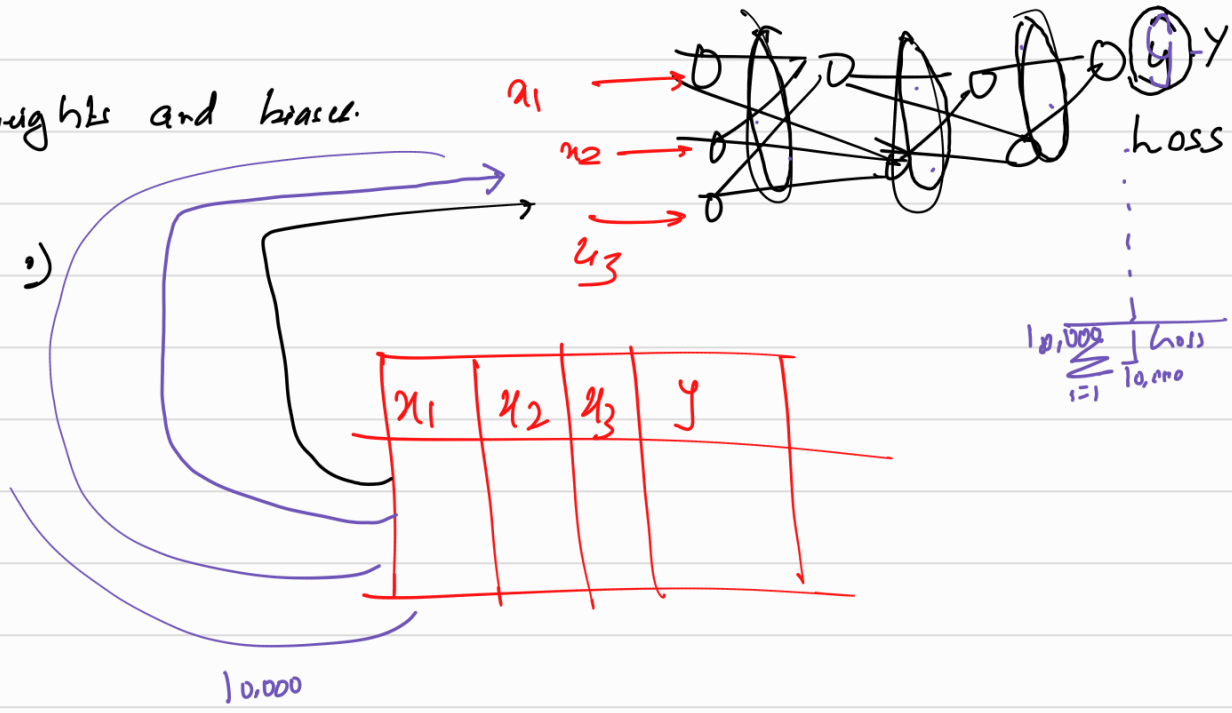
vs

$-\log(0.9) = 0.045$

$$\begin{matrix}0\\0\\0\end{matrix} \quad \begin{matrix}0\\0\\0\end{matrix} \rightarrow \text{Softmax, ohe}$$

log Loss

Random weights and biases.

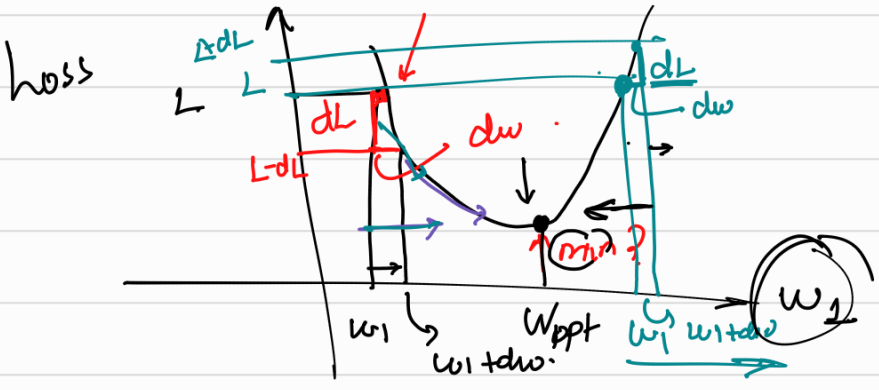


To minimise the loss

$$L = f(y_{pred}, y_{actual})$$

$$L = f((w, B, x), y_{actual})$$

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



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

which side we should move

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

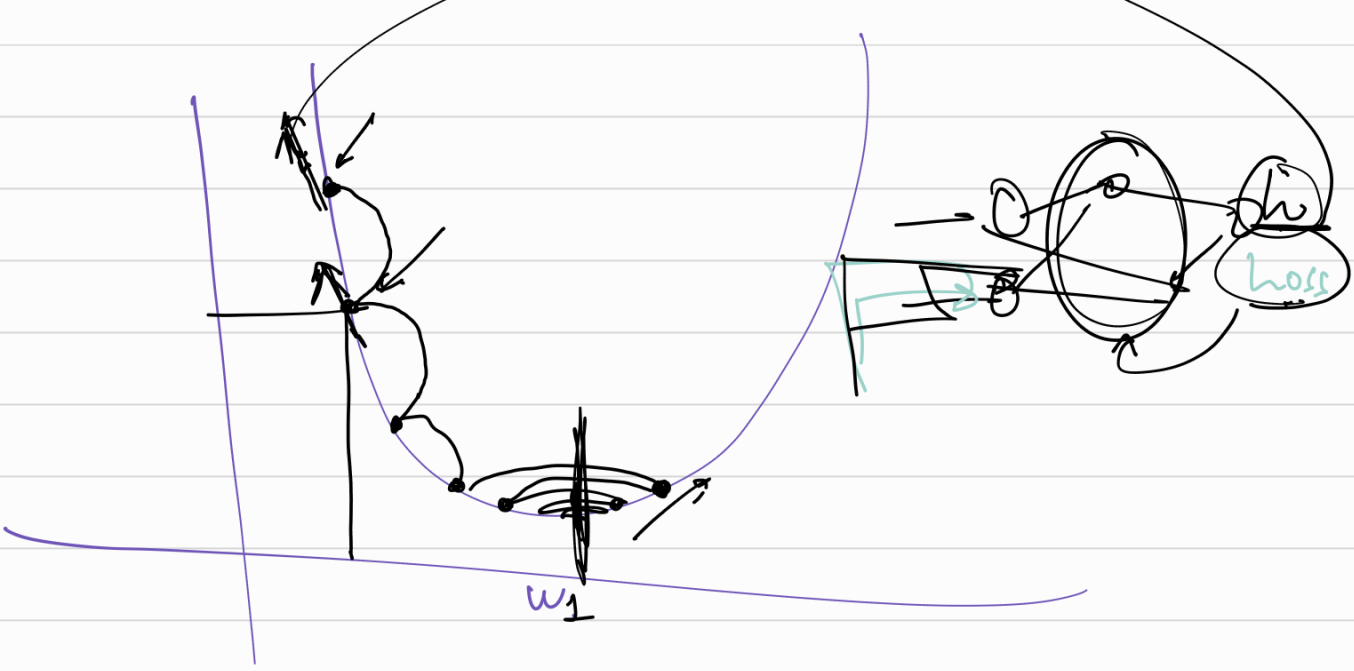
w = -ve.

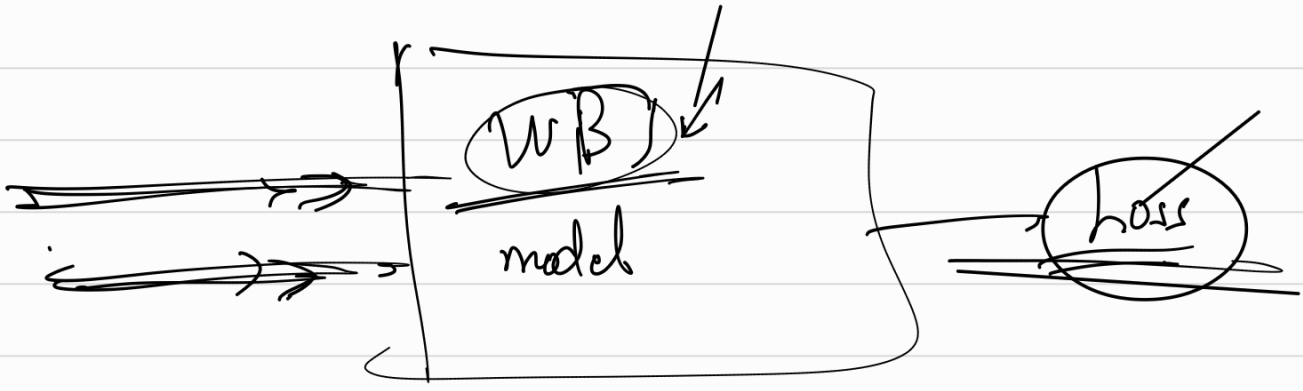
$$w_1(new) = w_1(old) - \alpha\left(\frac{dL}{dw_1}\right)$$

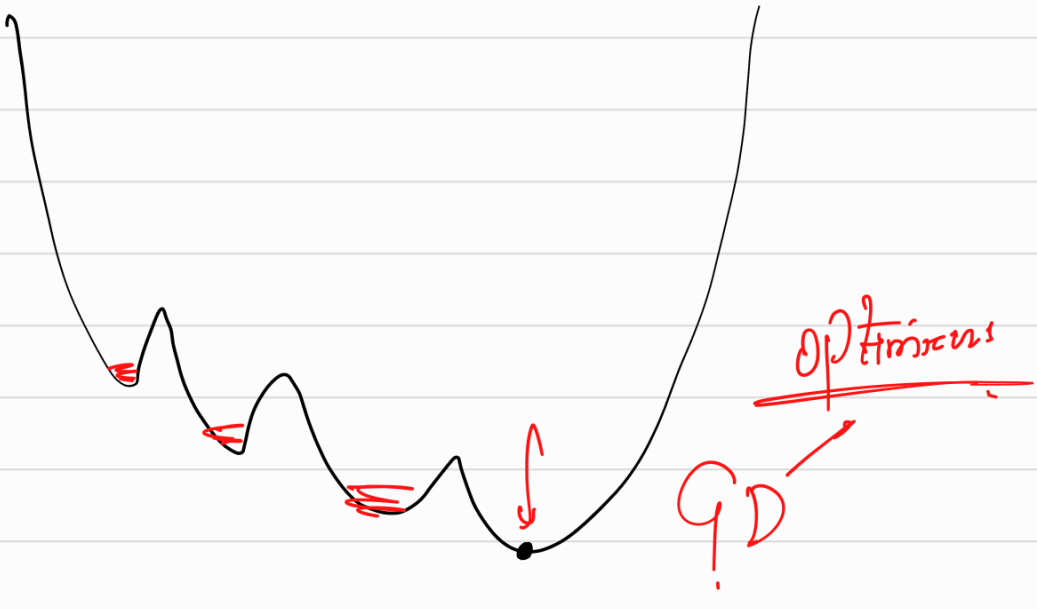
Hyp

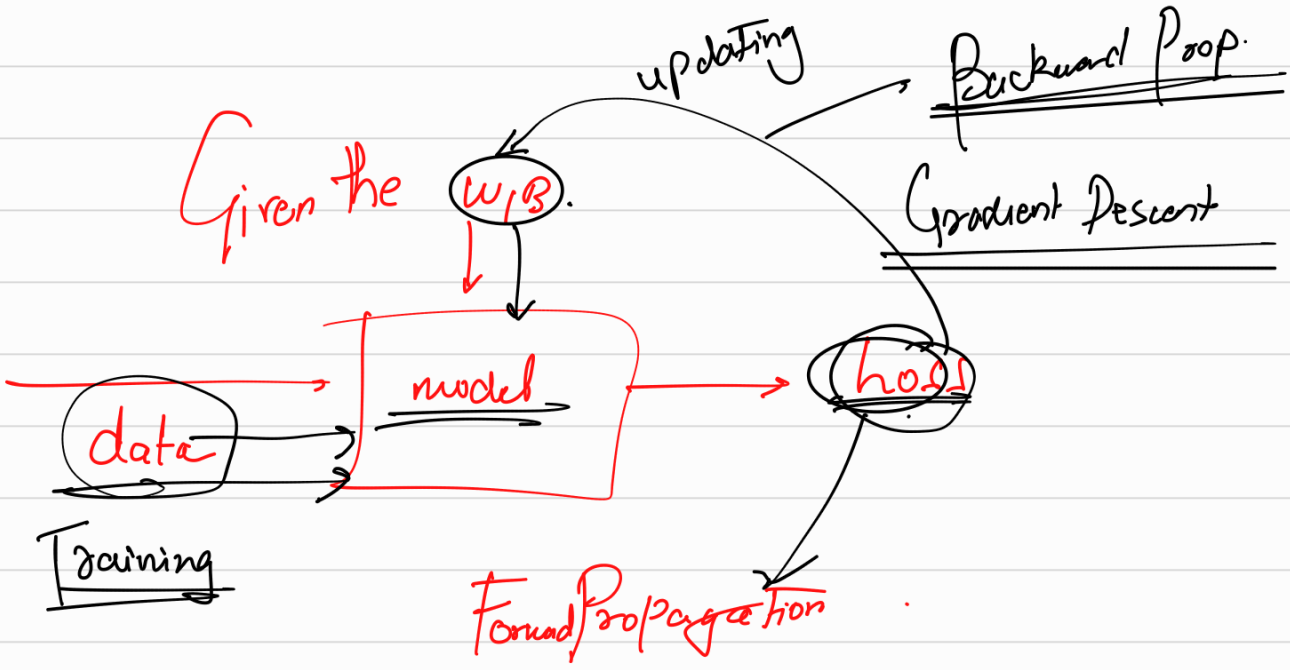
Partial

applicable for all weights and biases.


loss
w_1


WB
model
loss


optimizers
GD

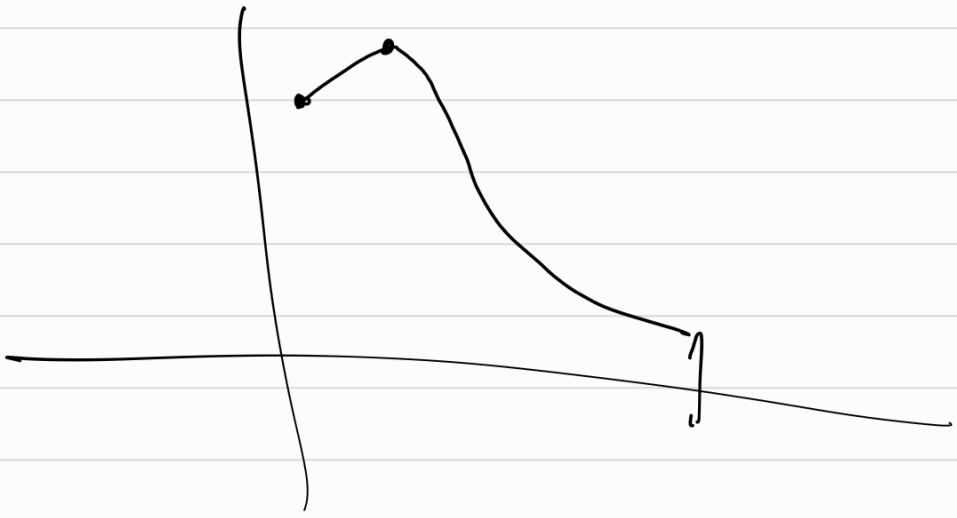


(1FP + 1BP) → epoch

(1FP, 1BP) → 1 epoch.

188 epoch

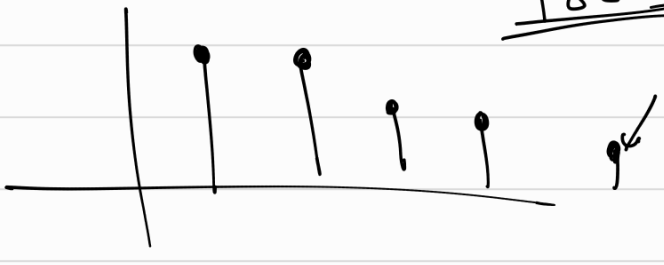
188 (1FP + 1BP)

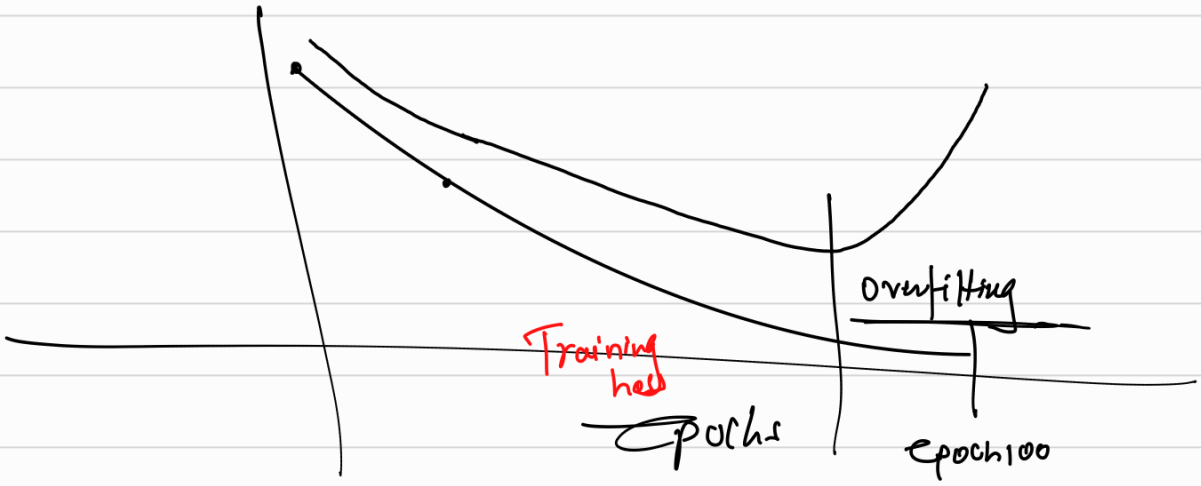


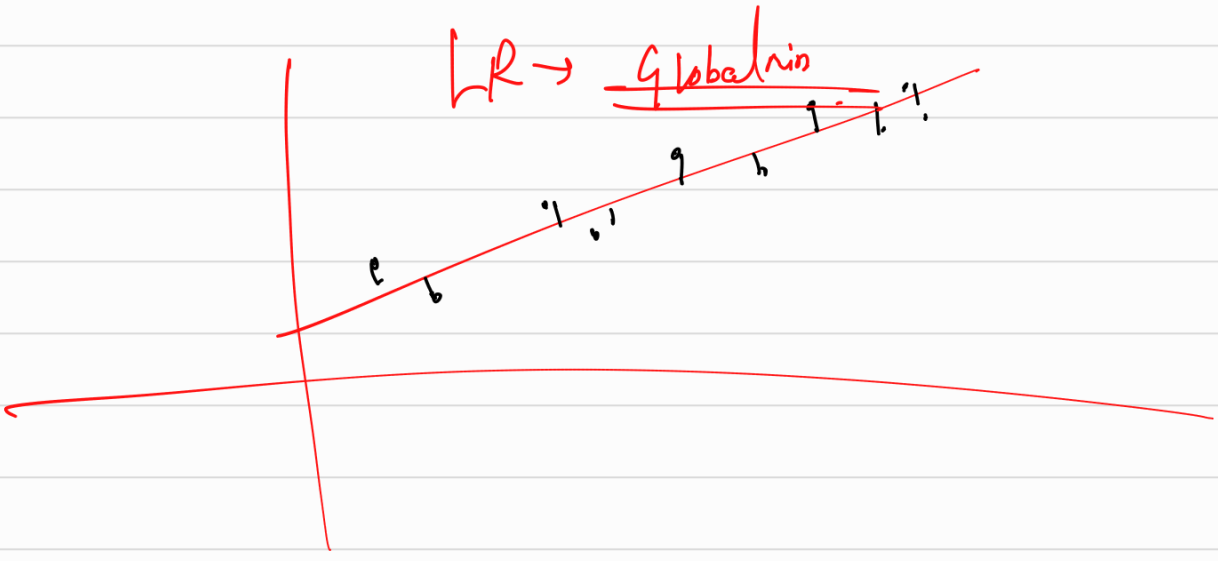
epoch 1 of 1

loss

188




Overfitting
Training
halt
Epochs
epoch100


LR → Global min