Neural Networks

- Artificial neural networks → Tabular
- Convolution neural network → images
- NLP (RNN, LSTM, Transformers) → text

ANN -

- why DL / what is DL → what is ANN
- What is a (Neuron)
- What is MLP
- Forward Prop / Backward Prop.
- Training steps of ANN

{ implement NN in a code (Tensorflow / Keras)

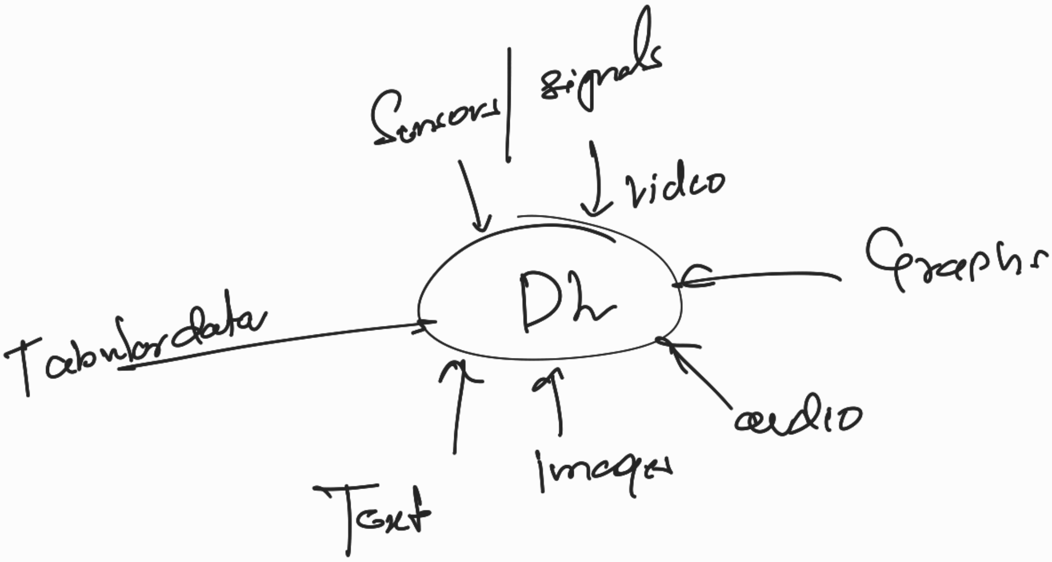
Dropout, Regularization, Batch normalization, Losses

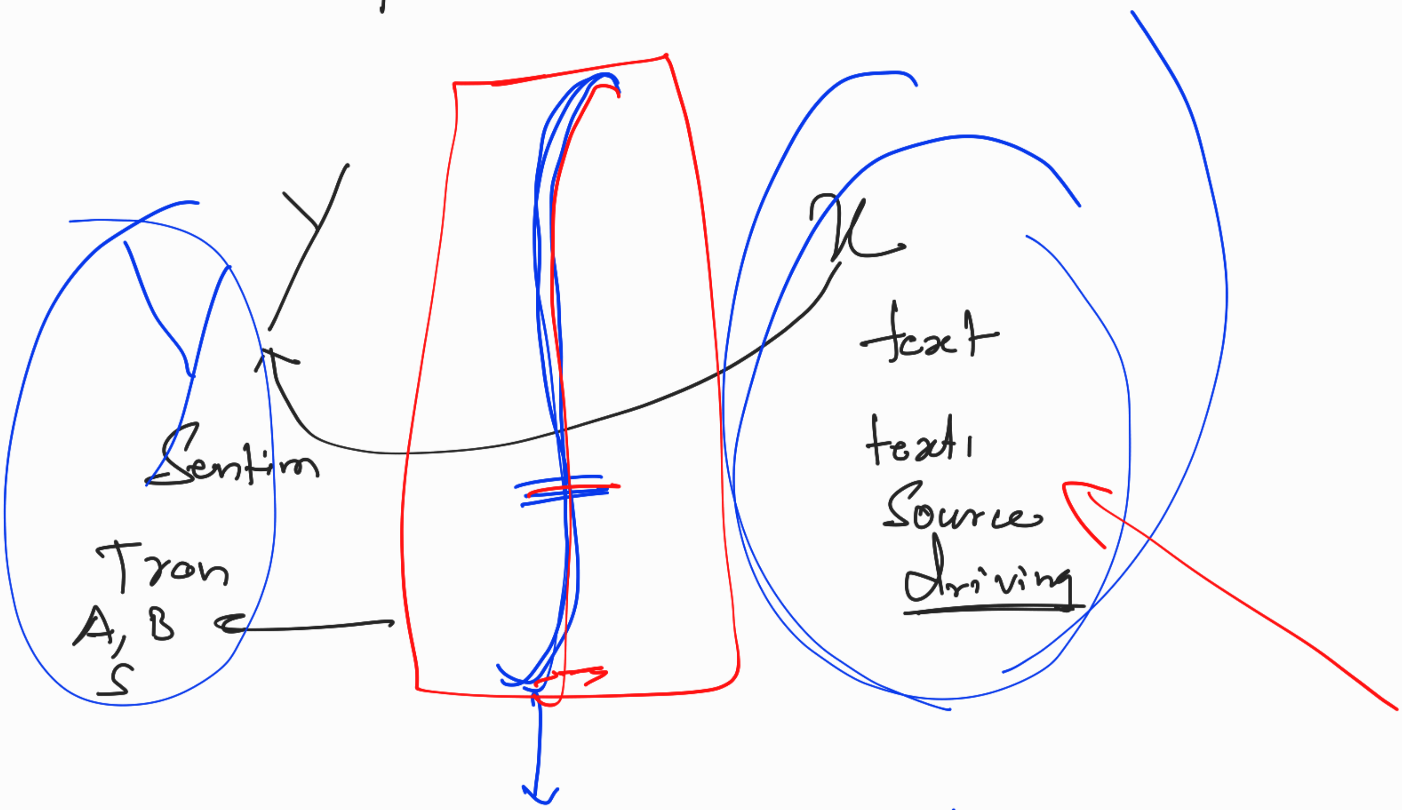
Callbacks, optimizers.

{ explainability / Denoising)
↳ Autoencoders

optimization using ANN.

Deep Learning is a field of study wherein we use parametric models which enables automated feature engineering.

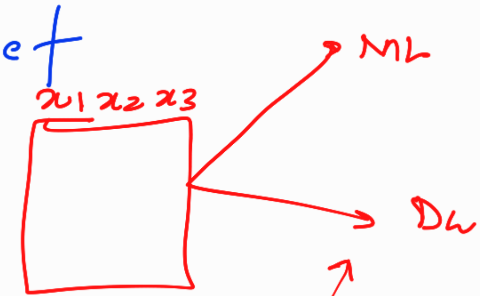


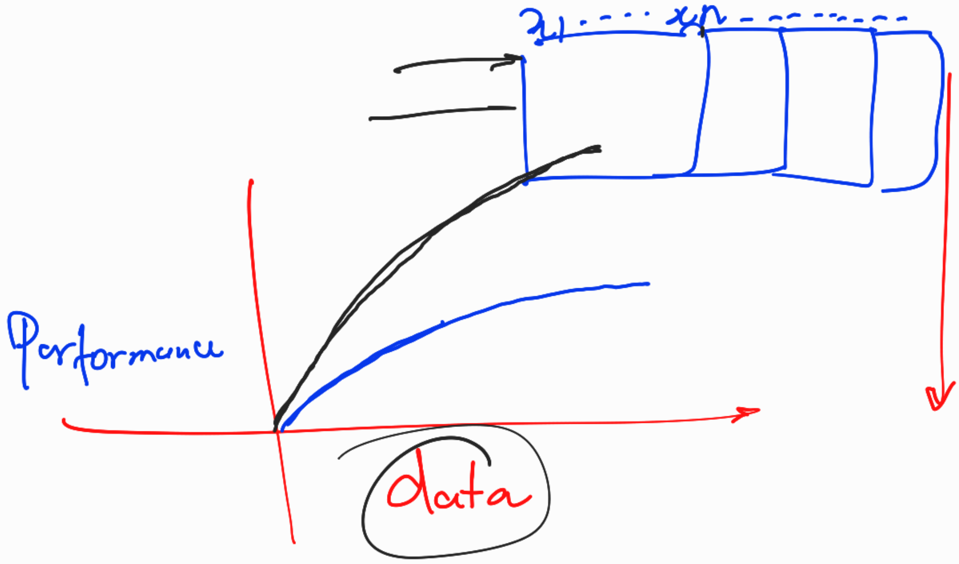


Architecture of the f

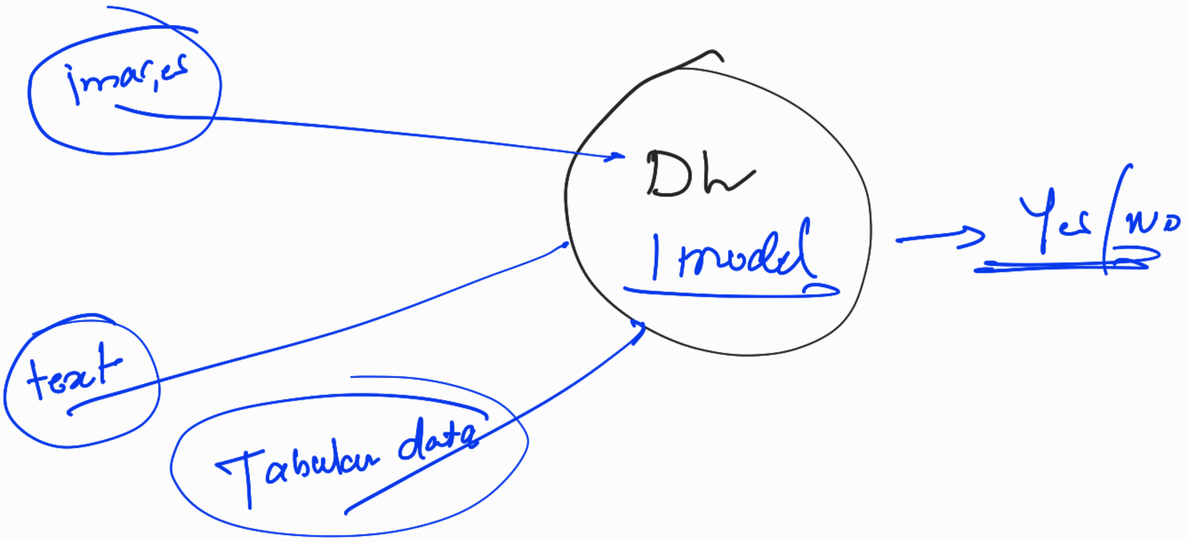
Tabular → ANN.
images → CNN.
graphs → GNN
text → Transformer

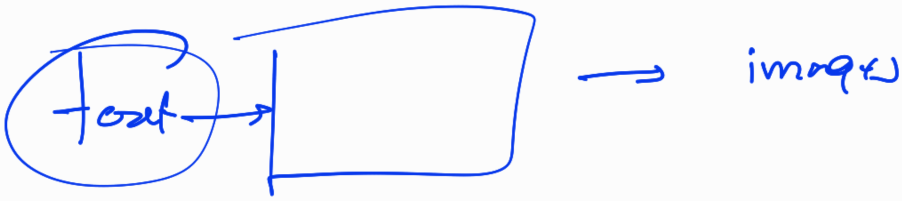
DL

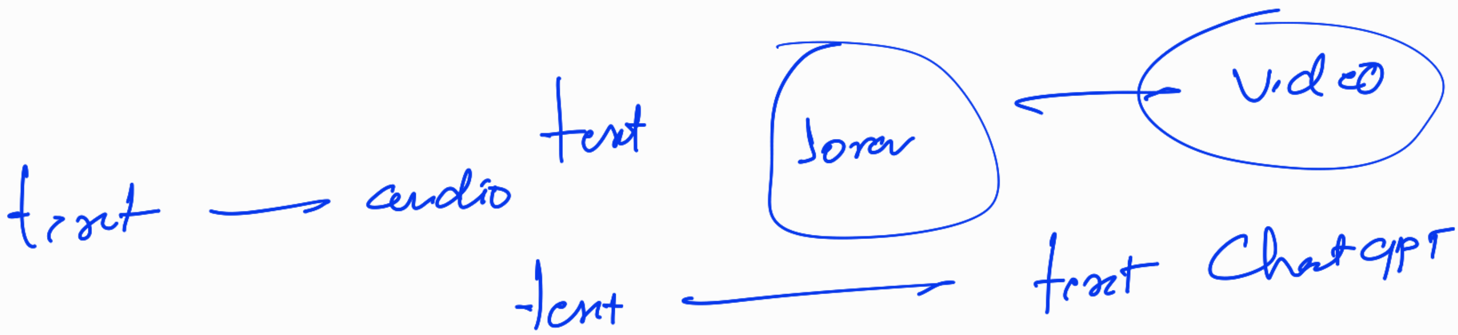




Multi-Modality

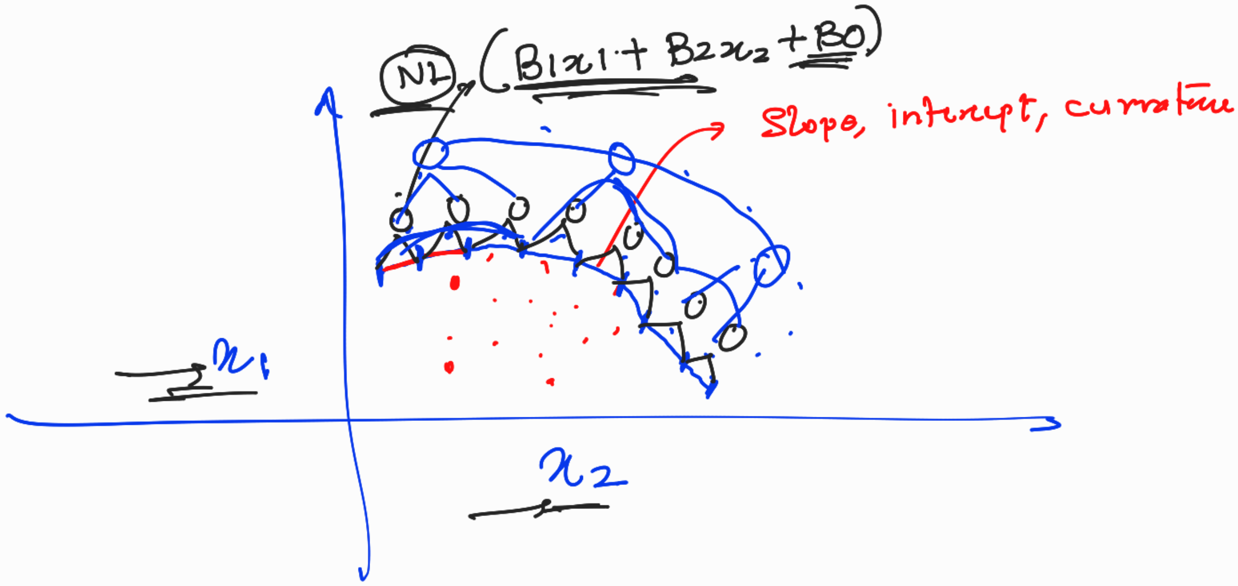




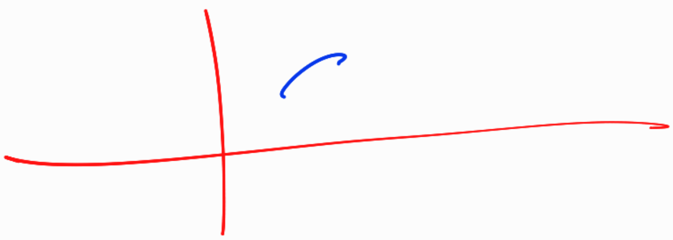


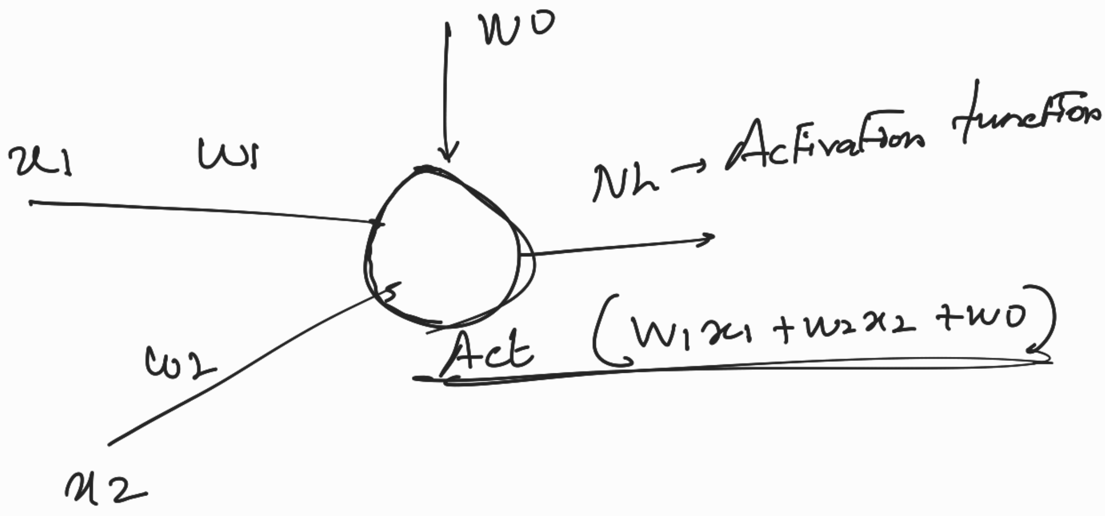
DL

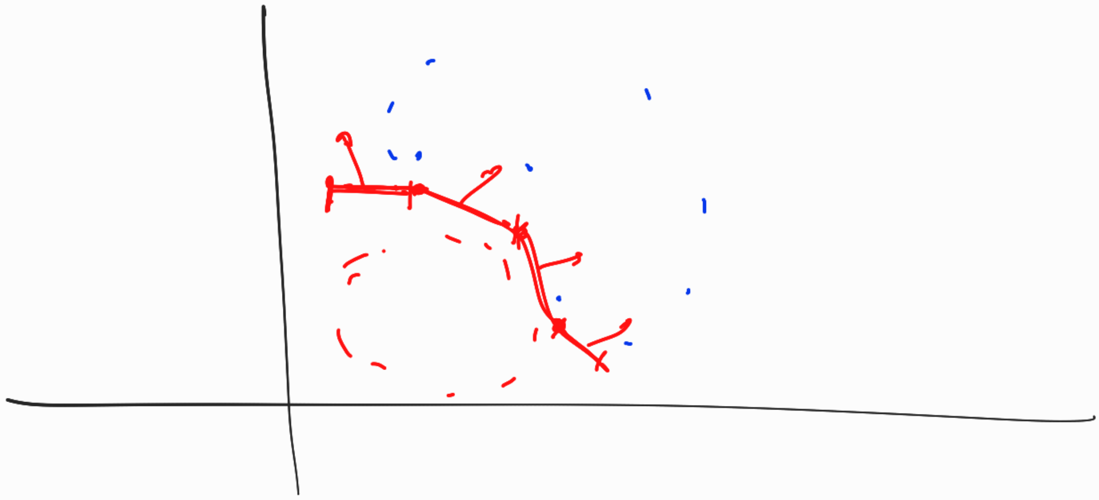
↳ Neurons

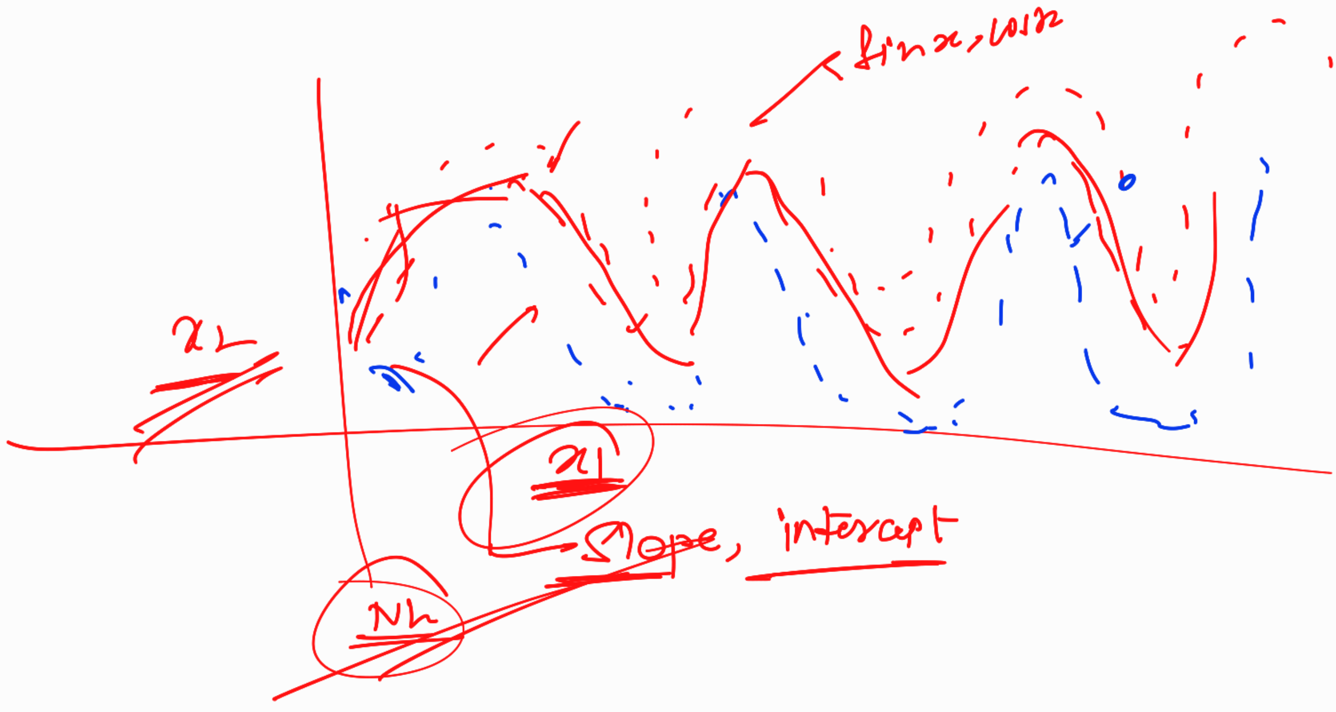


Can we use multiple "small" ^curvature^ 'logistic regressions' to create this blue boundary.

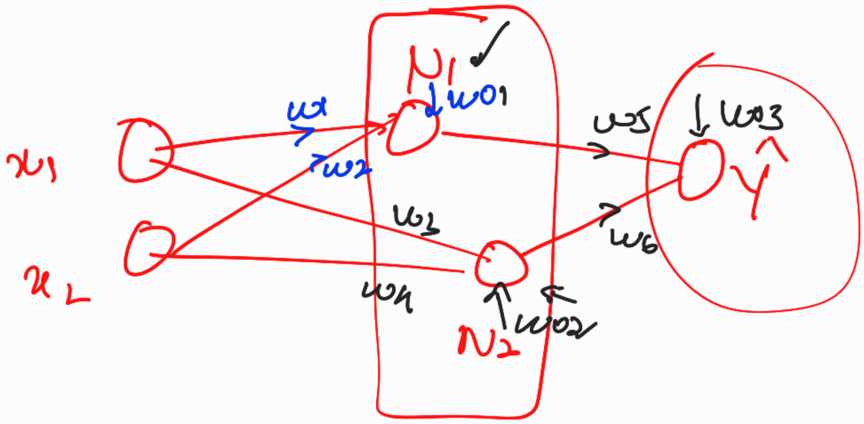








22:19 pm



$$N_1 = w_1 x_1 + w_2 x_2 + w_{01}$$

$$N_2 = w_3 x_1 + w_4 x_2 + w_{02}$$

$$\hat{y} = w_5 N_1 + w_6 N_2 + w_{03}$$

$$\hat{y} = w_5 (w_1 x_1 + w_2 x_2 + w_{01}) + w_6 (w_3 x_1 + w_4 x_2 + w_{02}) + w_{03}$$

$$\hat{y} = x_1 (w_5 w_1 + w_6 w_3) + x_2 (w_5 w_2 + w_6 w_4) + w_{01} w_5 + w_{02} w_6 + w_{03}$$

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

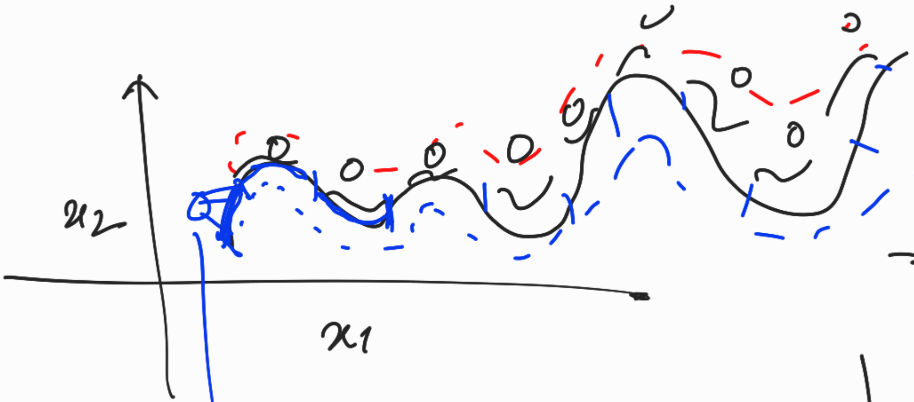
$$\boxed{\hat{y} = x_1 A + x_2 B + C}$$

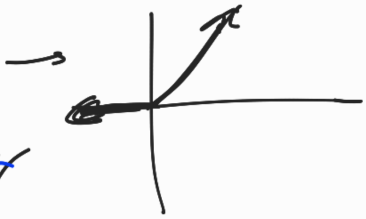
H|W →

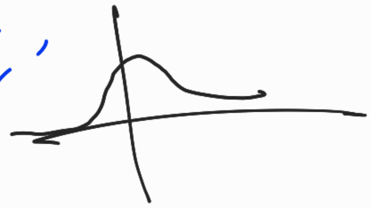
22/04/2024

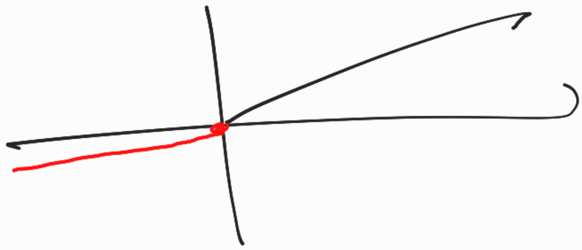
B → coefficient

NL → Activation





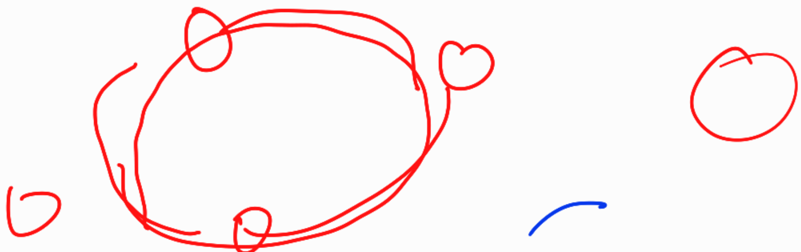


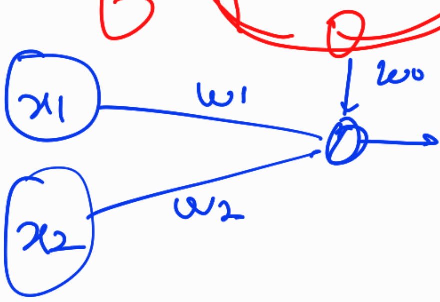


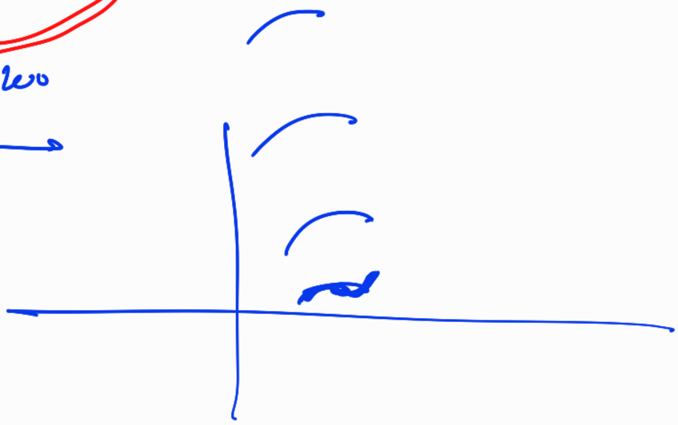
Slope

intercept

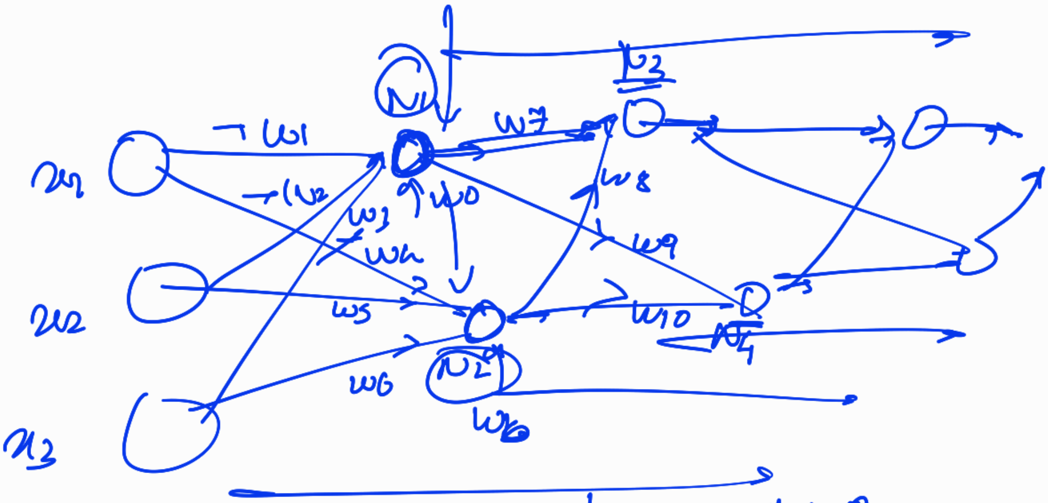
NL → Activation functions







$$\tanh(w_1 x_1 + w_2 x_2 + w_0)$$



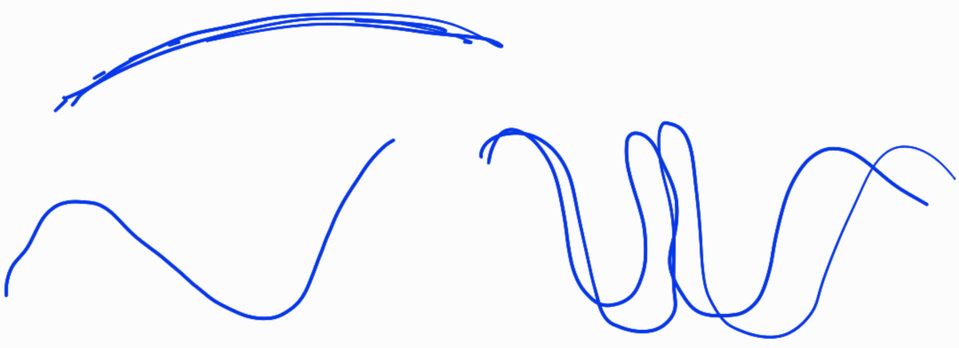
layers NN↑

$$N_1 = \tanh(w_1 x_1 + w_2 x_2 + w_3 x_3 + w_0)$$

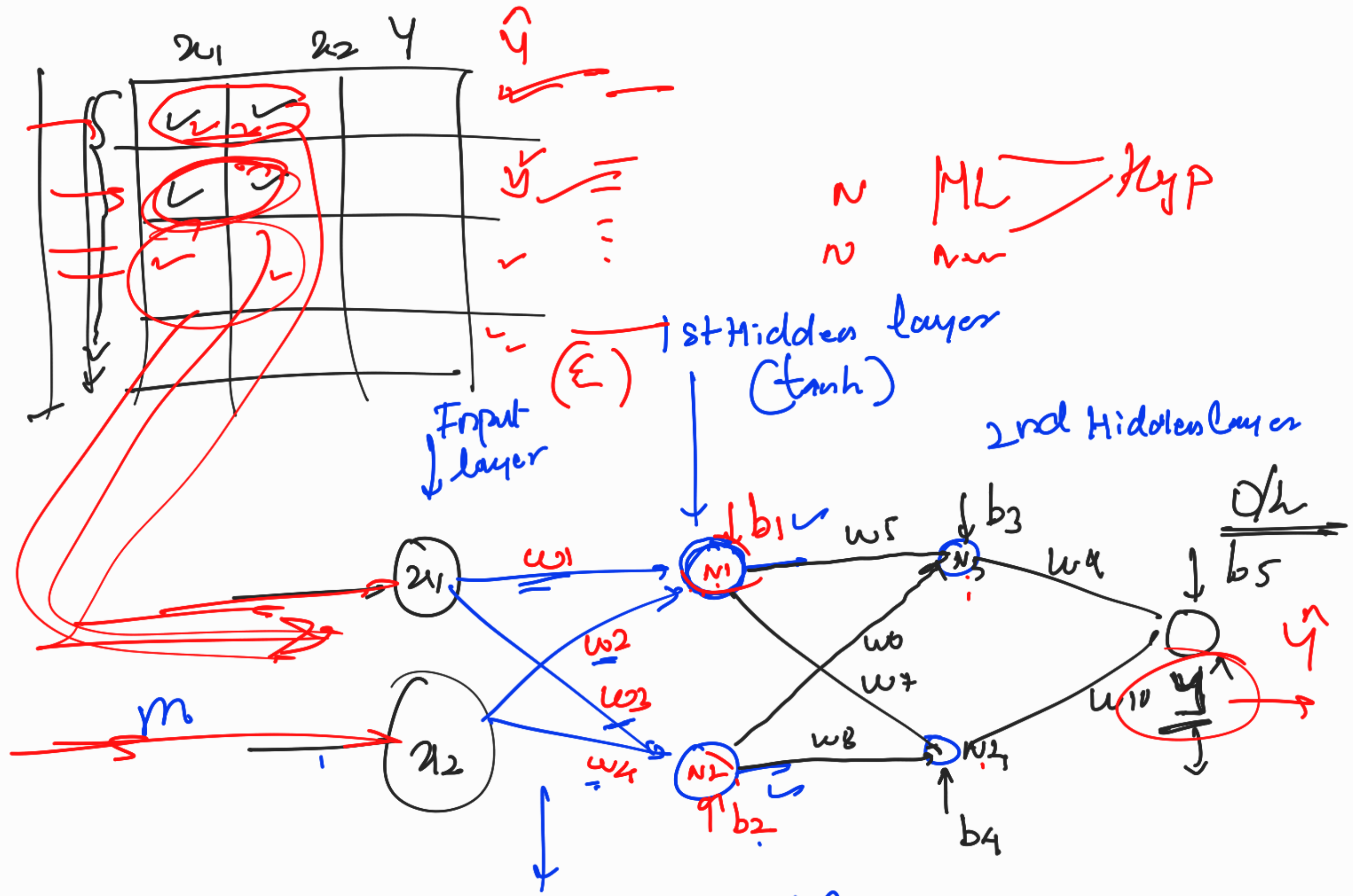
$$N_2 = \tanh(w_4 x_1 + w_5 x_2 + w_6 x_3 + w_b)$$

$$N_3 = \tanh(w_7 N_1 + w_8 \times N_2 + b)$$

$$N_4 = \tanh(w_9 N_1 + w_{10} N_2 + b)$$



Forward Propagation & Backward Propagation



Fully connected layers

Dense

Artificial neural networks

As soon as the DS introduces the first hidden layers by mentioning the number of neurons in the HL, immediately all connections are established and weights are assigned randomly.

activation function

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

$$N_2 = \tanh(w_3 x_1 + w_4 x_2 + b_2)$$

$$N_3 = \tanh(w_5 N_1 + w_6 N_2 + b_3)$$

$$N_4 = \tanh(w_7 N_1 + w_8 N_2 + b_4)$$

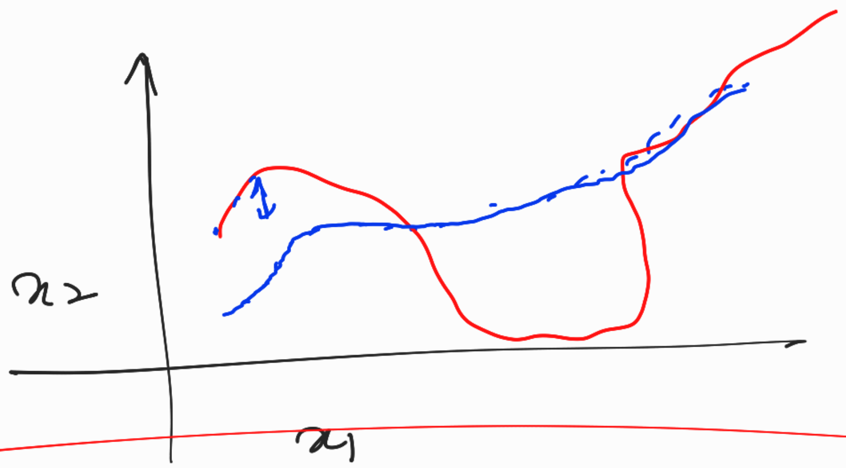
$$\hat{y} = (w_9 N_3 + w_{10} \times N_4 + b_5)$$

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

Forward Propogation

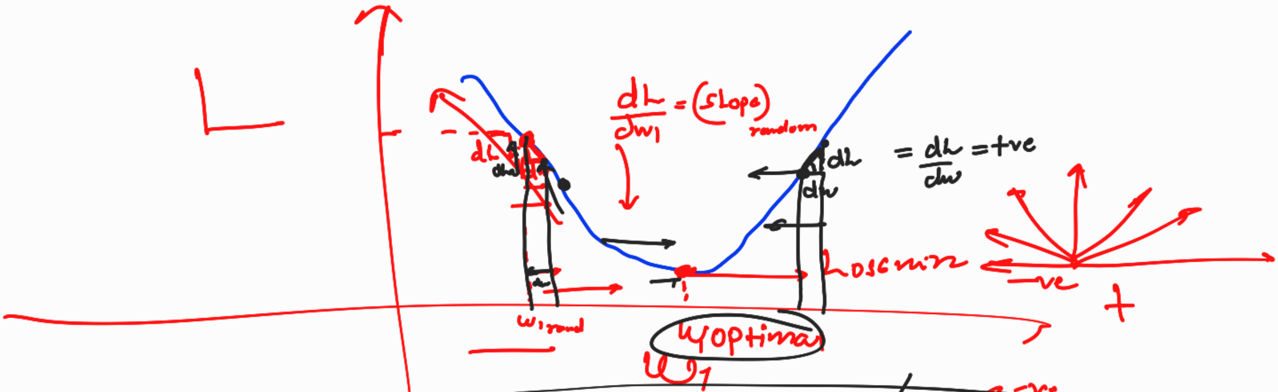
1st Forward Propogation

- Define the architecture of the NN (2HL, ...)
- The weights will be randomly assigned (dist).
- Define the loss function
- take all the sample data and calculate the loss.



- Minimise the loss by changing the weights
  Backward Propogation → Gradient Descent

$$L = f(\underline{w})$$

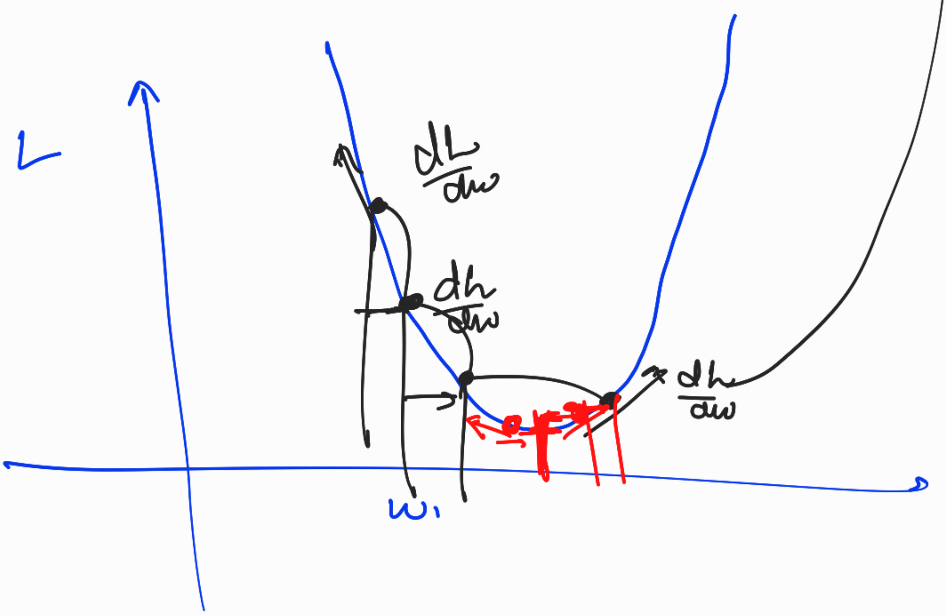


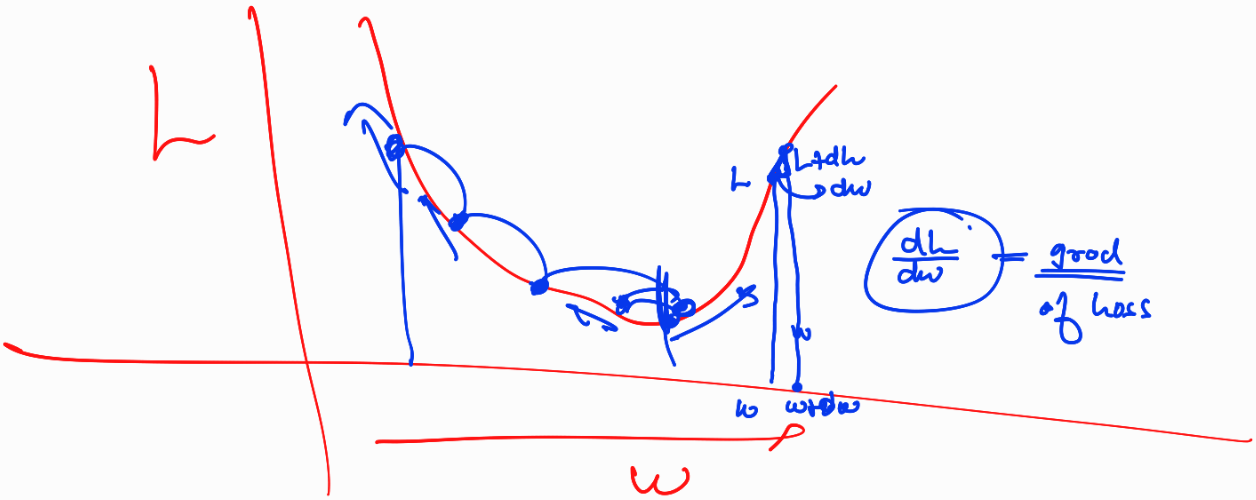
$$w_1(\text{next}) = w_1(\text{current}) - \alpha \frac{dL}{dw}$$

$$w_1(\text{next}) = w_1(\text{current}) - \alpha \frac{dL}{dw}$$

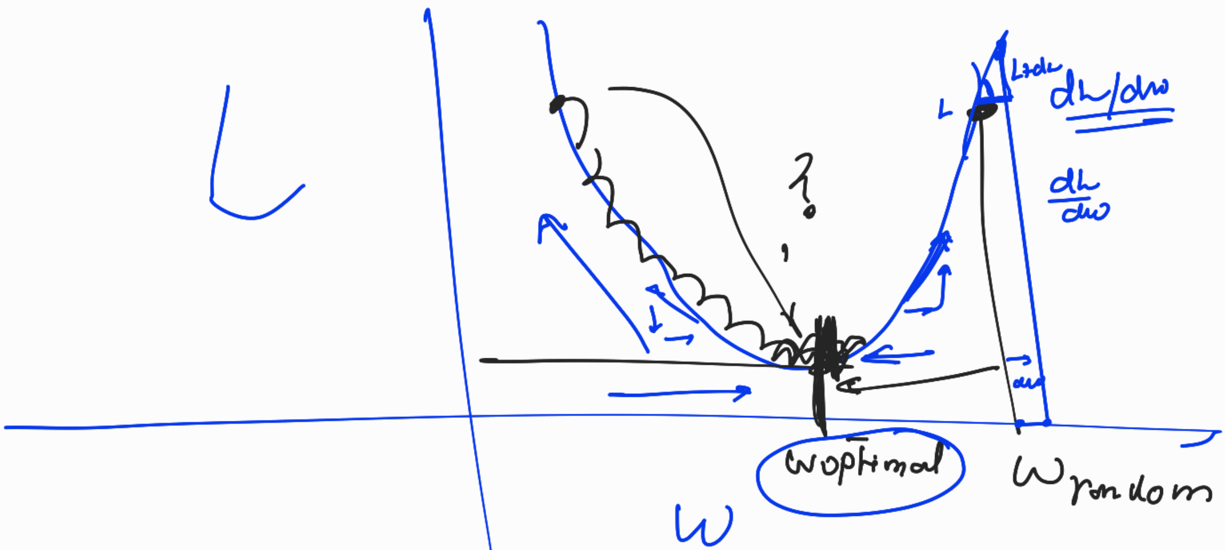
w_1 should move in the opp direction of the gradient.

$$\boxed{W_s(t+1) = w_s(t) - \alpha \frac{dL}{dw_s(t)}}$$





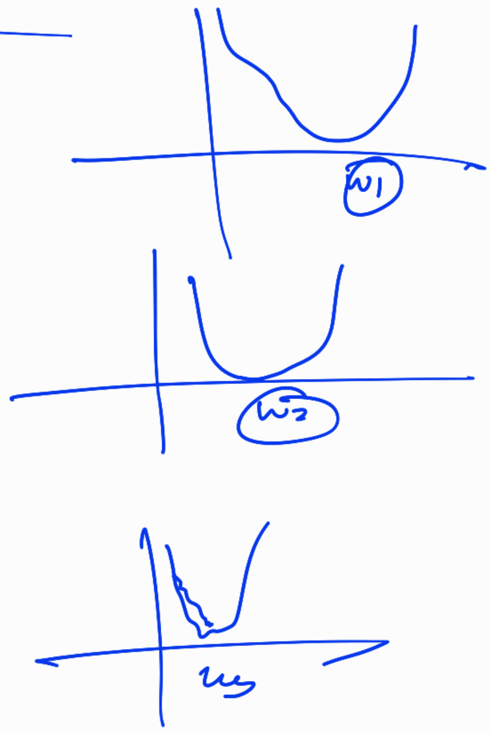
$$\left.\frac{dL}{dw}\right|_{w} = \text{Slope of the curve at}$$



$$w \propto \frac{dL}{dw}$$

$w_1, w_2, w_3, b_1, b_2, b_3$

$$\frac{\partial L}{\partial w_i}$$



Gradient Descent

Break. (22:20pm)

- Create the NN architecture and define the loss function.
- all weights are assigned randomly.
- Send all the (training data) through the architecture and calculate the loss for the entire data.

FP.

- given the loss, find out the derivative of the loss w.r.t all weights & biases.
- Perform gradient descent $(w(t+1)) = w_t - \alpha \frac{dL}{dw(t)}$

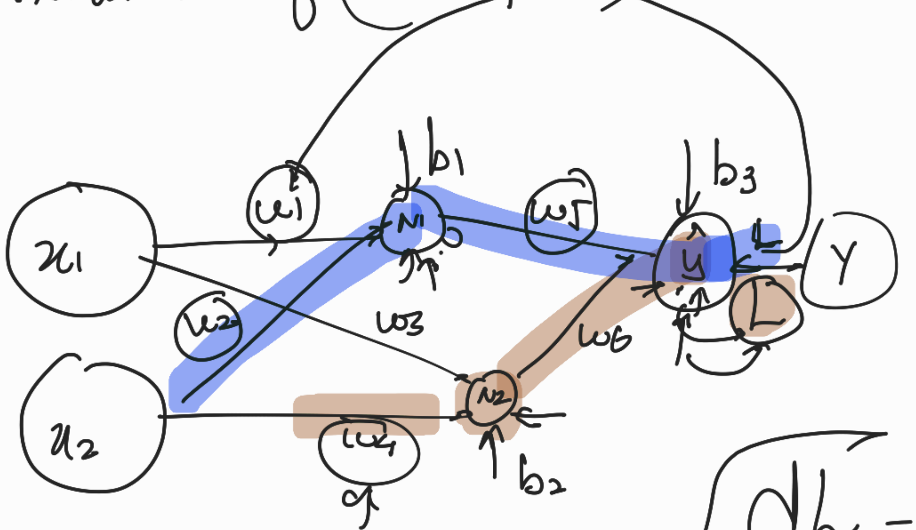
→ BP

⊙ update the weights)

⊙ Using the update weights, goto step 3.

⊙ The model's loss doesnt reduce further (saturates).

⊙ Each iteration of (1 FP + 1 BP) → 1 epoch.



$N_1 = w_1 x_1 + w_2 x_2 + b_1$

$N_2 = w_3 x_i + w_4 x_2 + b_2$

$\hat{y} = w_5 N_1 + w_6 N_2 + b_3$

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

$$\frac{dL}{dw_5} = \frac{dL}{d\hat{y}} \times \frac{d\hat{y}}{dw_5}$$

$$\frac{dL}{dw_6} = \frac{dL}{d\hat{y}} \times \frac{d\hat{y}}{dw_6}$$

$$\frac{d(x^2)}{dx} = 2x$$

$$\frac{dL}{\partial w_1} \rightarrow \frac{\partial L}{\partial w_2}, \frac{\partial L}{\partial w_3} \dots \frac{dL}{dw_6}, \frac{dL}{\partial b_1}, \frac{\partial L}{\partial b_2}, \frac{\partial L}{\partial b_3}$$

$$\left[\sum_{i=1}^{n} \frac{1}{n} 2 (y_i - \hat{y}_i) x - 1\right] [w_6] \times (x_{2i})$$

$\frac{\partial L}{\partial w_1}$

$$\frac{dL}{dw_4} = \frac{dL}{d\hat{y}} \times \frac{d\hat{y}}{dN_2} \times \frac{dN_2}{dw_4}$$

$$\frac{\partial L}{dw_2} = \frac{dL}{d\hat{y}} \times \frac{d\hat{y}}{dN_1} \times \frac{dN_1}{dw_2}$$

$$y = 5x$$

$$\frac{\partial y}{\partial x} = 5$$

$$y = 5x$$

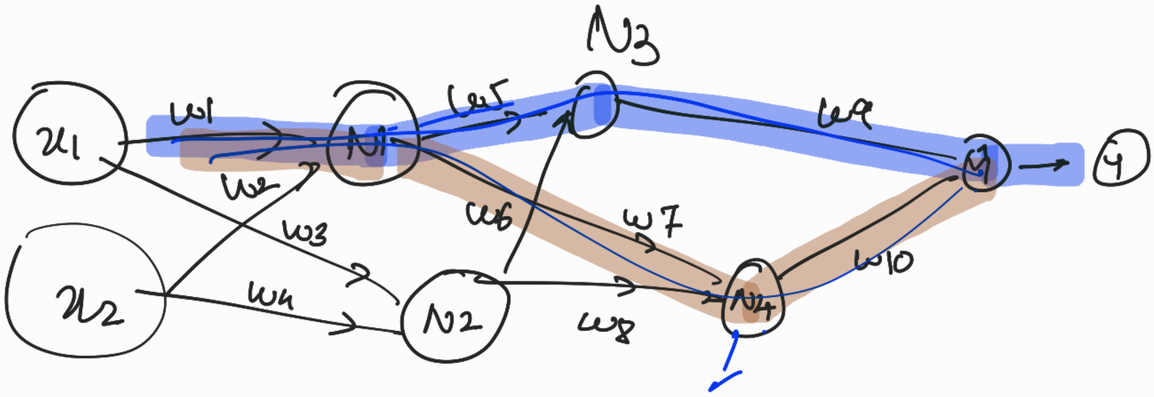
$$x = 5z - 1$$

$$x = 5$$

$$\frac{\partial y}{\partial z} = \frac{\partial y}{\partial x} \times \frac{dx}{dz}$$ → Chain Rule

$$1$$

$$\frac{\partial L}{\partial w} \quad \frac{\partial y}{\partial z}$$

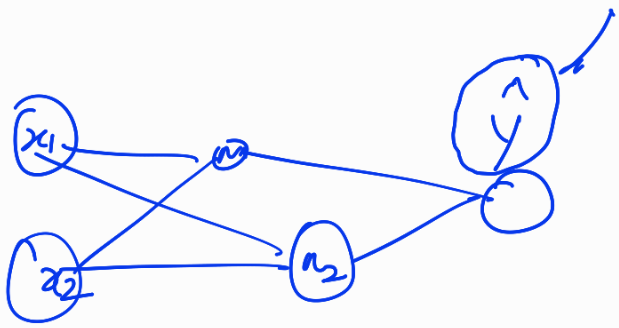


$$\frac{dL}{dw_1} = \frac{dL}{d\hat{y}} \times \frac{d\hat{y}}{dN_3} \times \frac{dN_3}{dN_1} \times \frac{dN_1}{dw_1}$$

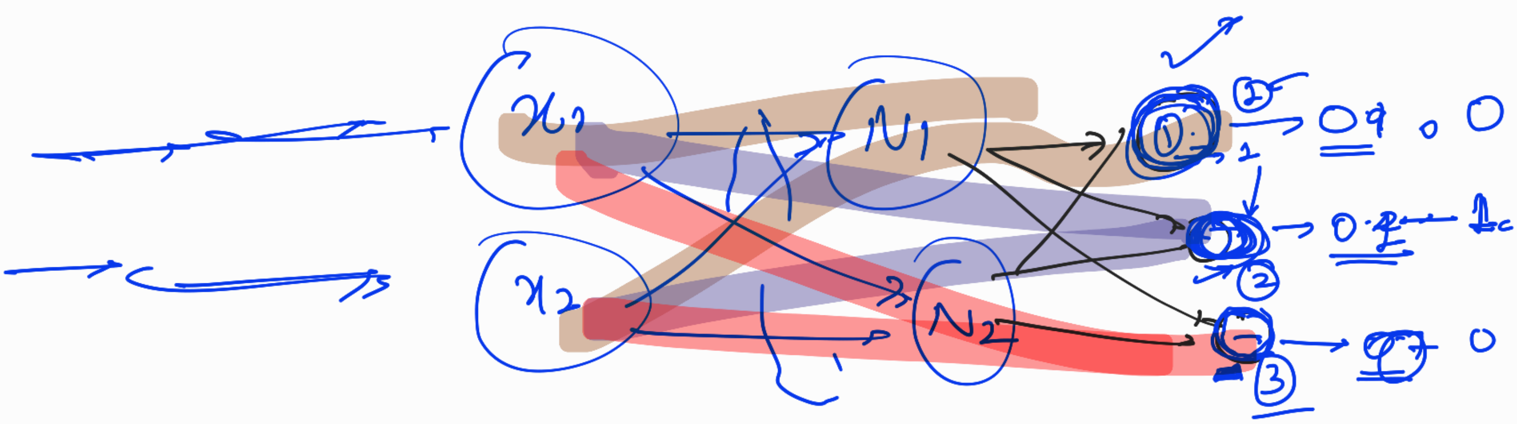
$$+$$

$$\frac{dL}{d\hat{y}} \times \frac{d\hat{y}}{dN_4} \times \frac{dN_4}{dN_1} \times \frac{dN_1}{dw_1}$$

x_1	x_2	y	1	2	3
✓	✓	1	1	0	0
✓	✓	2	0	1	0
✓	✓	3	0	0	1



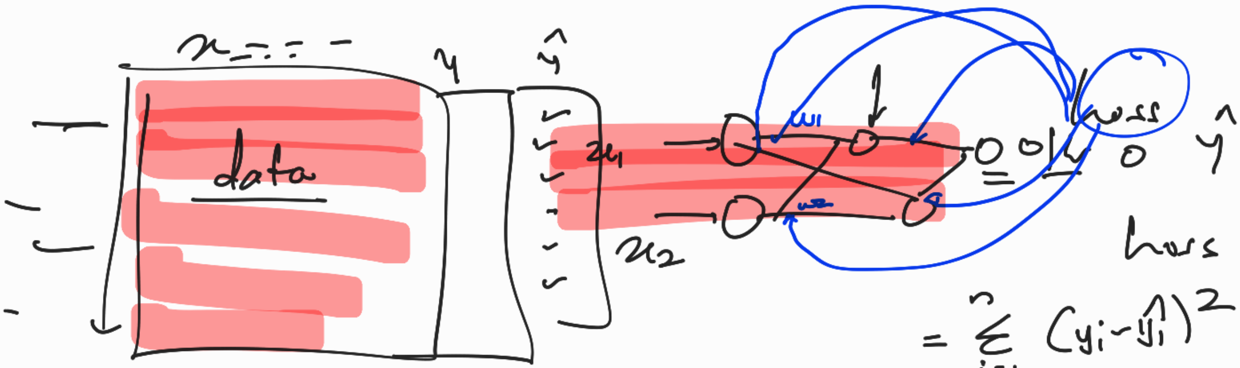
y



$$\begin{array}{ccc} j=1 & 2 & 3 \\ 1 & 0 & 0 \\ 0.8 & 0.1 & 0.1 \end{array}$$

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{3} y\log\hat{y} = 1\times\log 0.8 + 0\times\log 0.1 + 0\times\log 0.8$$

$\boxed{ABCD} \rightarrow$ 20 mins



loss

$$= \sum_{i=1}^{n} (y_i-\hat{y}_i)^2$$

$$\rightarrow \sum_{i=1}^{n} y_i \log \hat{y}_i$$

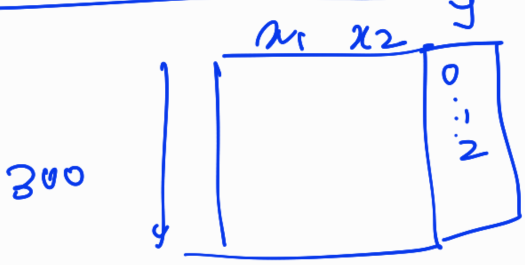
$$\frac{dL}{dw_1} \cdot \frac{dL}{dw_2} \cdot \frac{dL}{db_1} \ldots$$

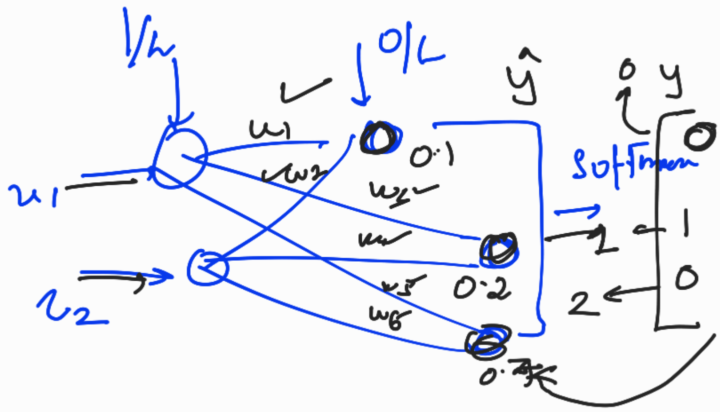
$$\boxed{w_i(t+1) = w_i(t) - \alpha\frac{dL}{dw_i(t)}} \quad ?$$

$$L = f(y, \hat{y})$$

gra...

Code





Loss = Categorical cross entropy

$$y = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

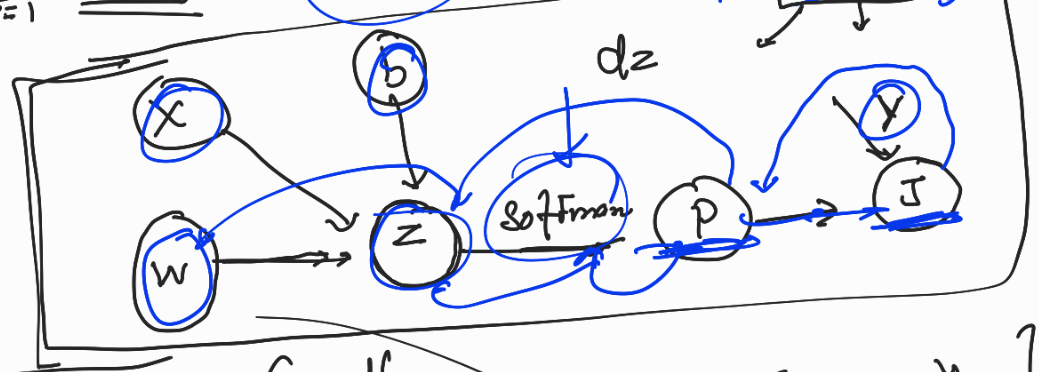
$$\hat{y} = \begin{bmatrix} 0.2 & 0.4 & 0.84 \end{bmatrix}$$

0.4

$$\sum_{j=1}^{3} y \log \hat{y}$$

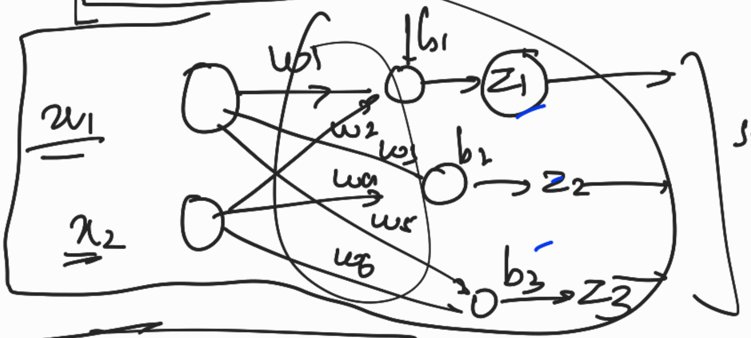
Scaler

$$\frac{dJ}{dw} = \frac{dJ}{dp} \times \frac{dp}{dz} \times \frac{dz}{dw}$$



$\frac{dJ}{dw}$

dJ/db



$$\begin{bmatrix} p_1 & 0 \\ p_2 & 1 \\ p_3 & 0 \end{bmatrix}$$

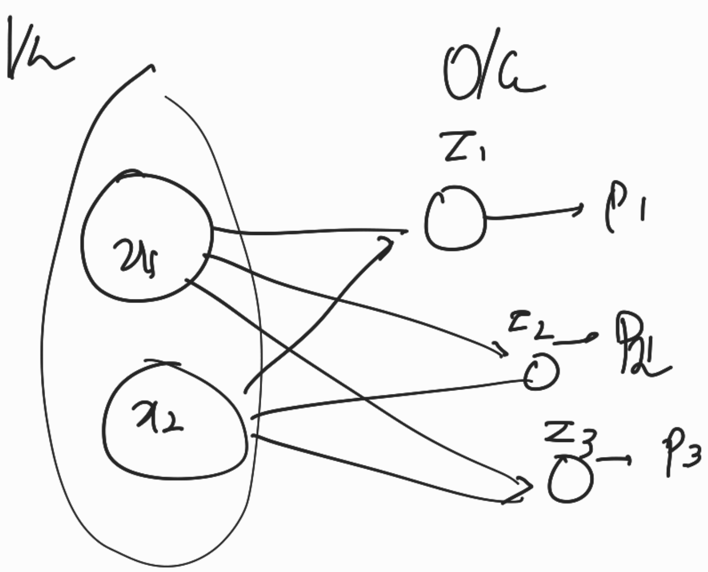
$$\boxed{Z = wx + b}$$

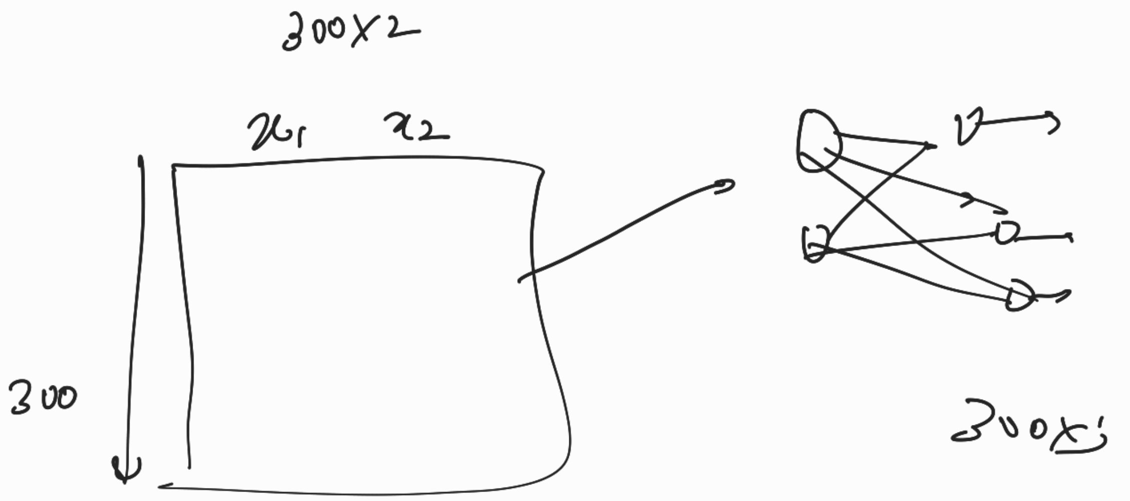
$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} =$$

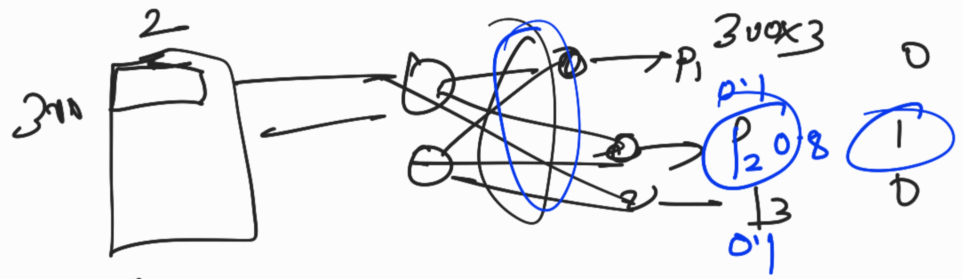
$x =$ 300 × 2

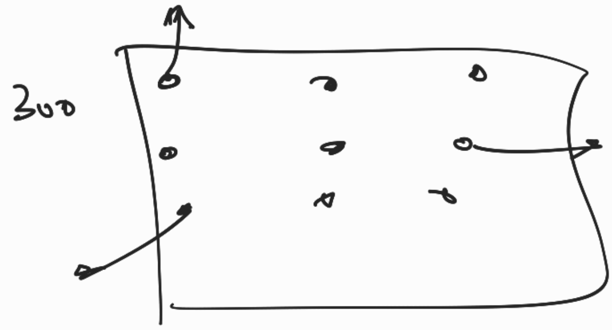
$$w = \begin{bmatrix} w_1 & w_2 \\ w_3 & w_4 \\ w_5 & w_6 \end{bmatrix} \quad 3 \times 2$$

$$Z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$$









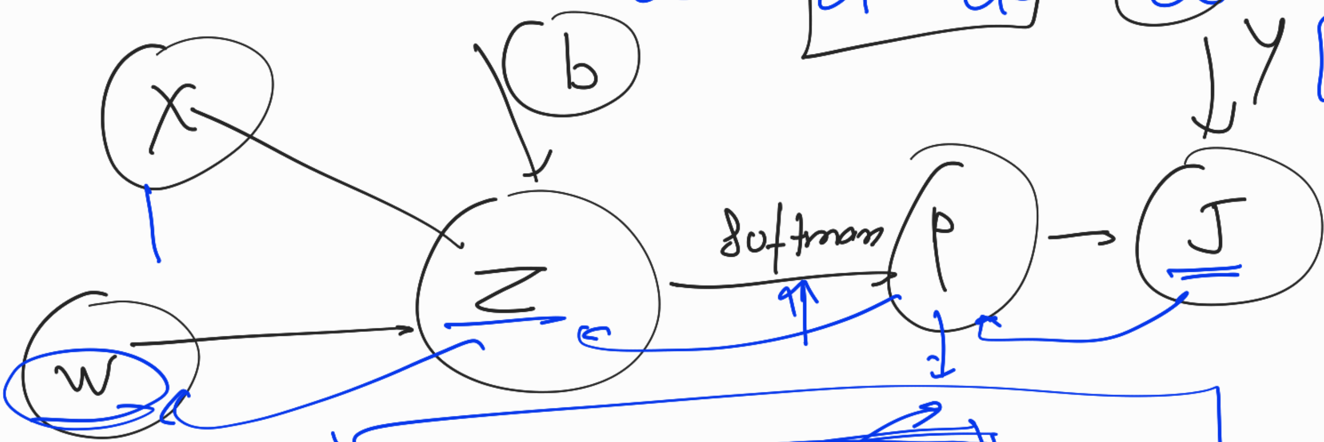
$$Z = wx + b$$

$$\frac{dz}{db} = 1$$

$$\frac{\partial J}{\partial b} = \boxed{\frac{\partial J}{\partial p} \times \frac{dp}{dz}} \times \frac{dz}{db}$$

dz

Y $[1\ 0\ 0]$



$$\boxed{\frac{dJ}{dw}} = \boxed{\frac{\partial J}{\partial p} \times \frac{dp}{dz}} \times \frac{dz}{dw}.$$

$$Z = wx + b$$

$$\frac{dz}{dw} = x$$

Calculus

$$-\, dz = \begin{array}{ccc} p_1 & p_2 & p_3 \\ \hline \end{array} \quad [\,p_1,\ p_2 - 1,\ p_3\,] \leftarrow \begin{array}{c} 2,\ y \\ [0,\ 1,\ 0] \end{array}$$

$$\frac{d x^2}{dx} = 2x$$

$$Z = w x + b$$

$$\frac{dz}{dw} = x$$

$$\frac{dJ}{dw} = dz \times \frac{dz}{dw}$$

$$dJ/dw = dz \times x$$

$dz = prob$

300 × 3

$dz[\,range(m),\ y\,] -= 1$

300x3

		p_1	p_2	p_3	y
300	1	0.2-1	0.3	0.5	0
	2	0.7	0.2-1	0.1	1
	3	0.8	0.1	0.1-1	2

y
0
1
2

$dJ/dw = dz \times dZ/dw$

$dJ/d_B = dz \times dZ/db$

3

$Z =$

300×3

300

3

300

2×300

300

3

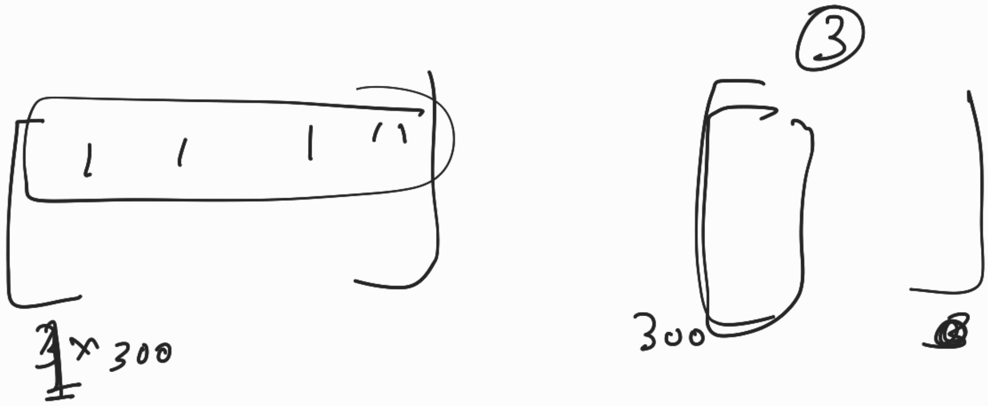
300

3

1×3

300

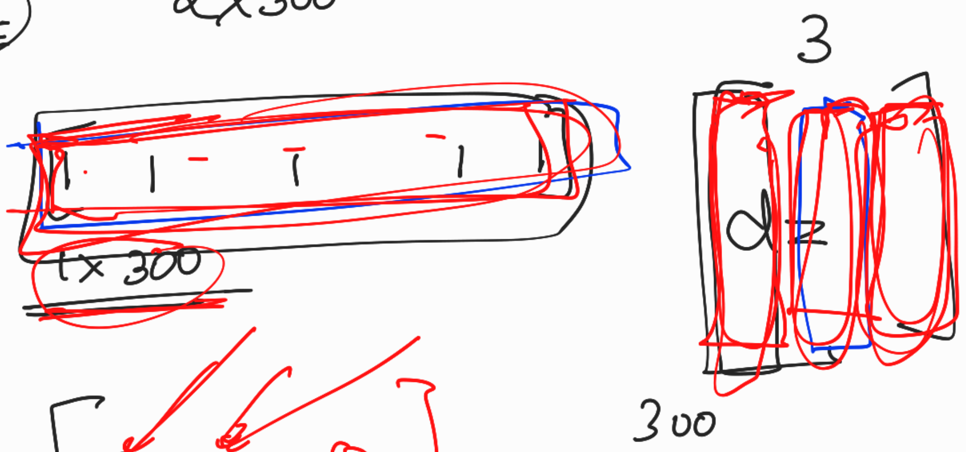
3



$$\frac{dz}{dw} = x^{T} \quad 2 \times 300$$

$$\frac{dz}{db} = 1 \qquad 2 \times 300$$

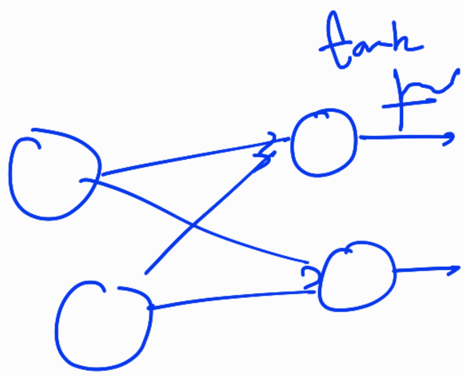
$$\frac{dz}{db} =$$



$$\frac{dz}{db} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$$

$\frac{dz}{db} =$ np.sum dz (axis=0

Activation functions:

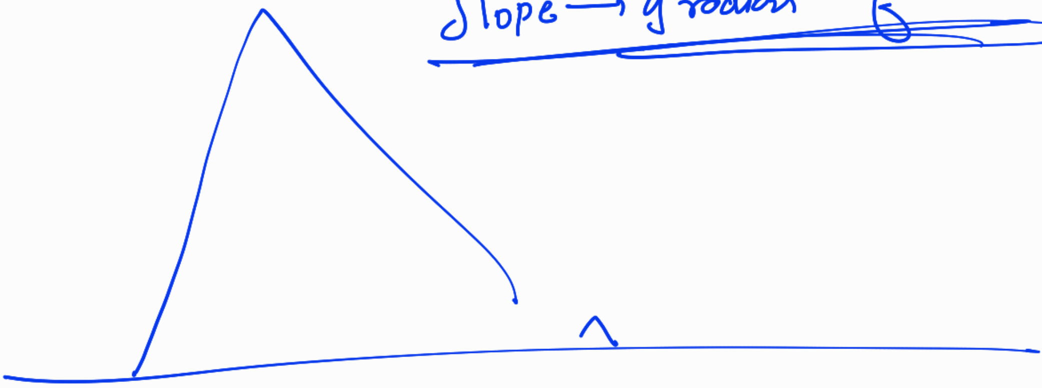


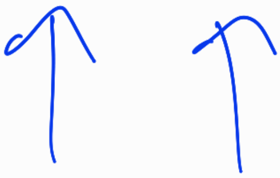
Relu

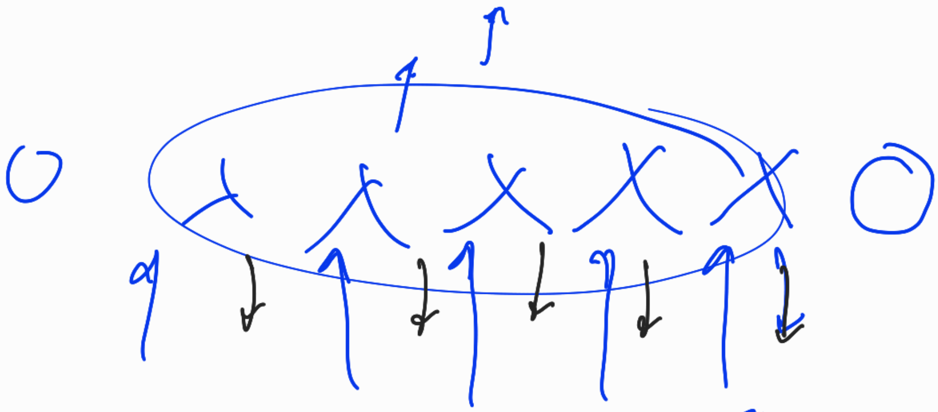
Continuous / differentiable

Computationally expensive.

Slope → gradient of act







$$w_{i+1} = w_i - \alpha \, dJ/dw$$

≤ 0