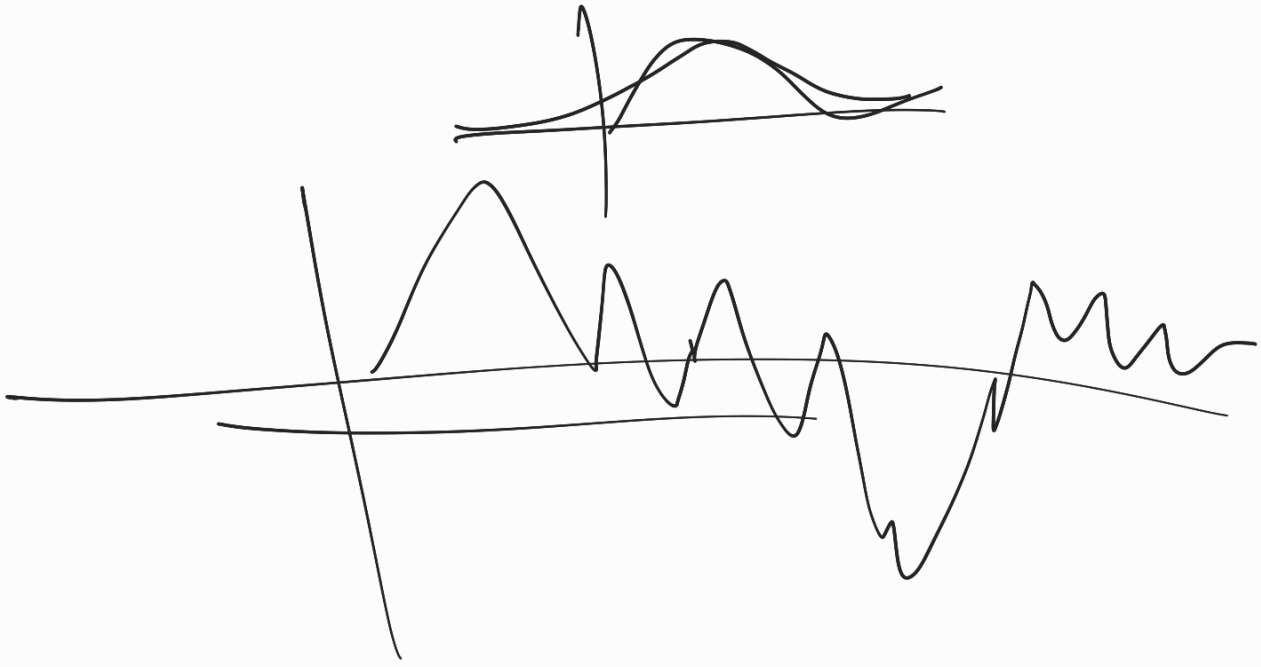


26/4/2024

- Activation functions
- How does a FP, BP look like for a 1HL NN. (activation functions)

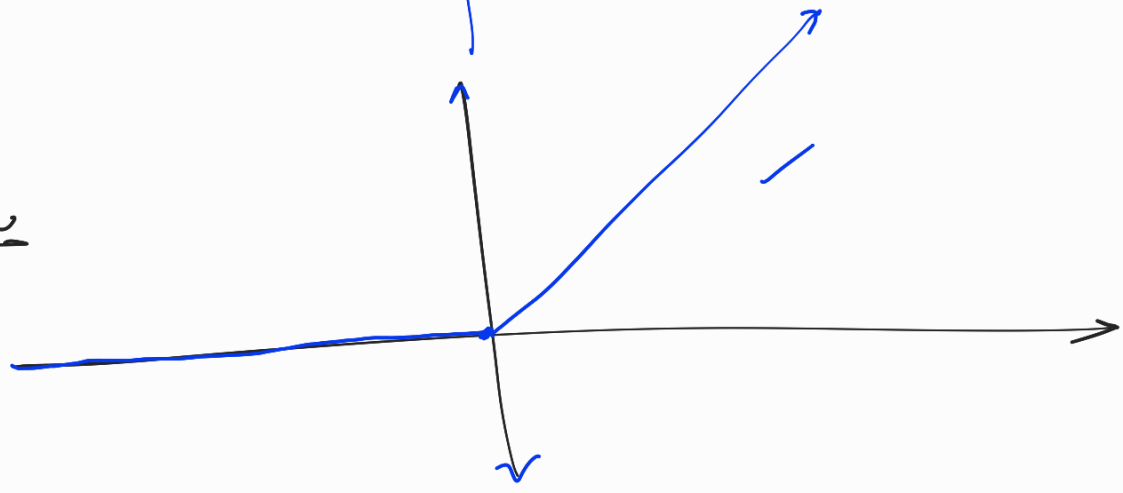
- Activation functions → functions which are applied at the Neuron output in HL / OL to specifically bring in Non-linearity in the system.

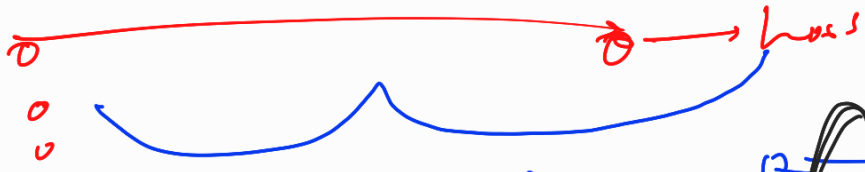
HL – Activation → Non-linearity

OL → Activation → Restrict the output specific

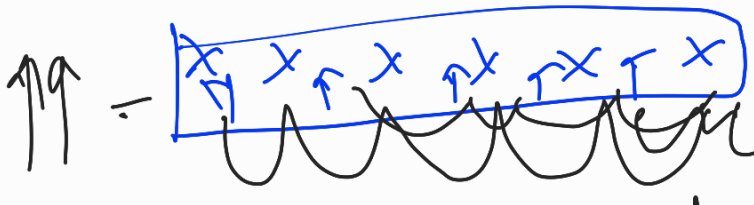
Problem

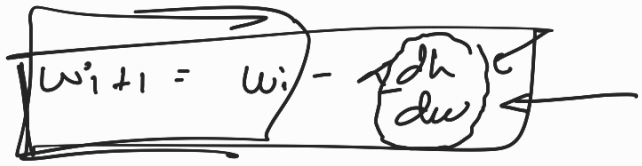
Relu

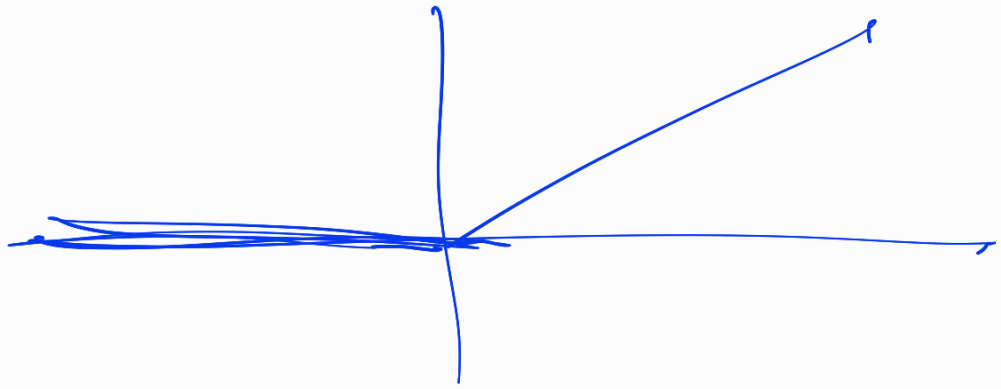


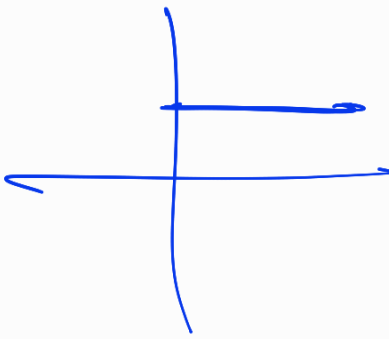


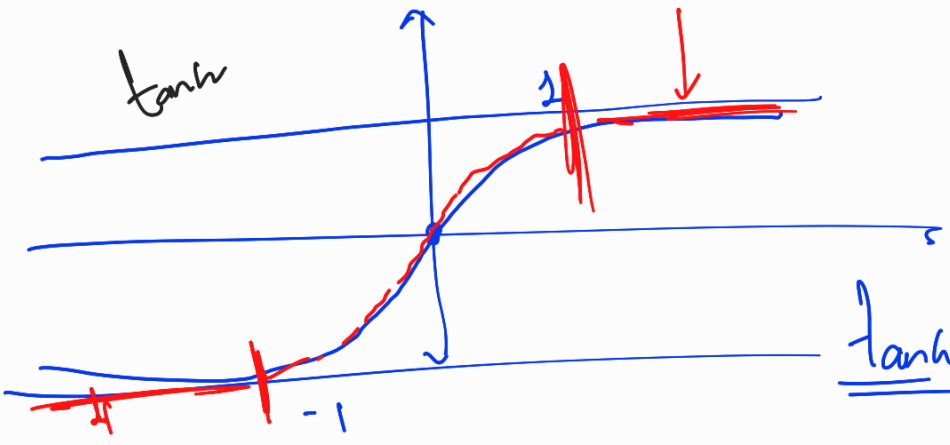
Chain Rule

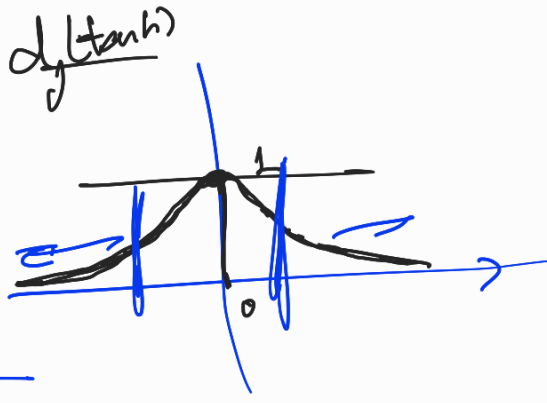


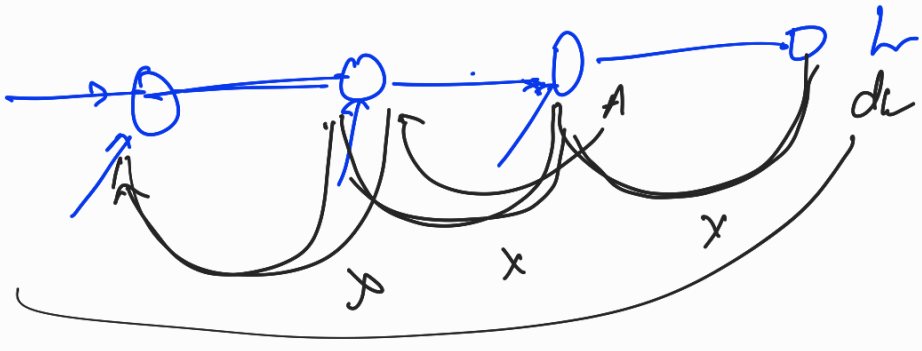






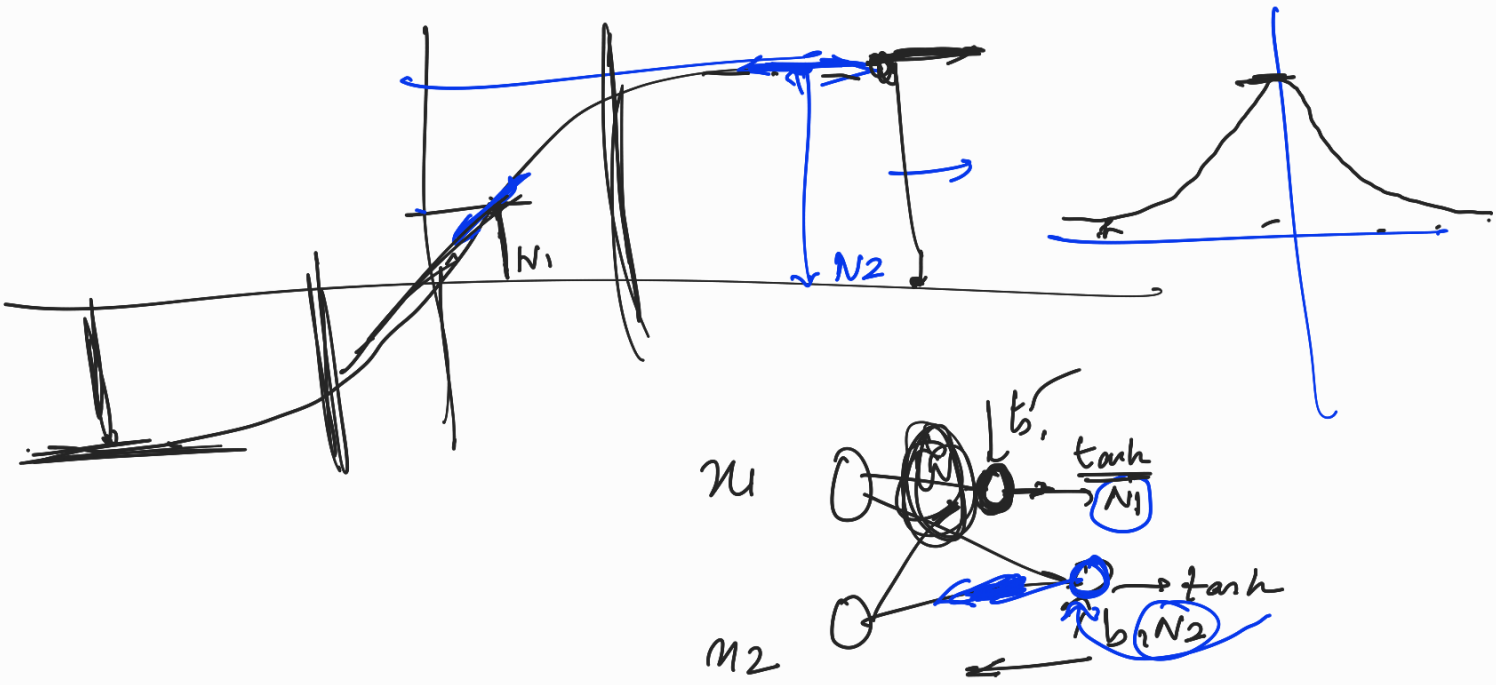






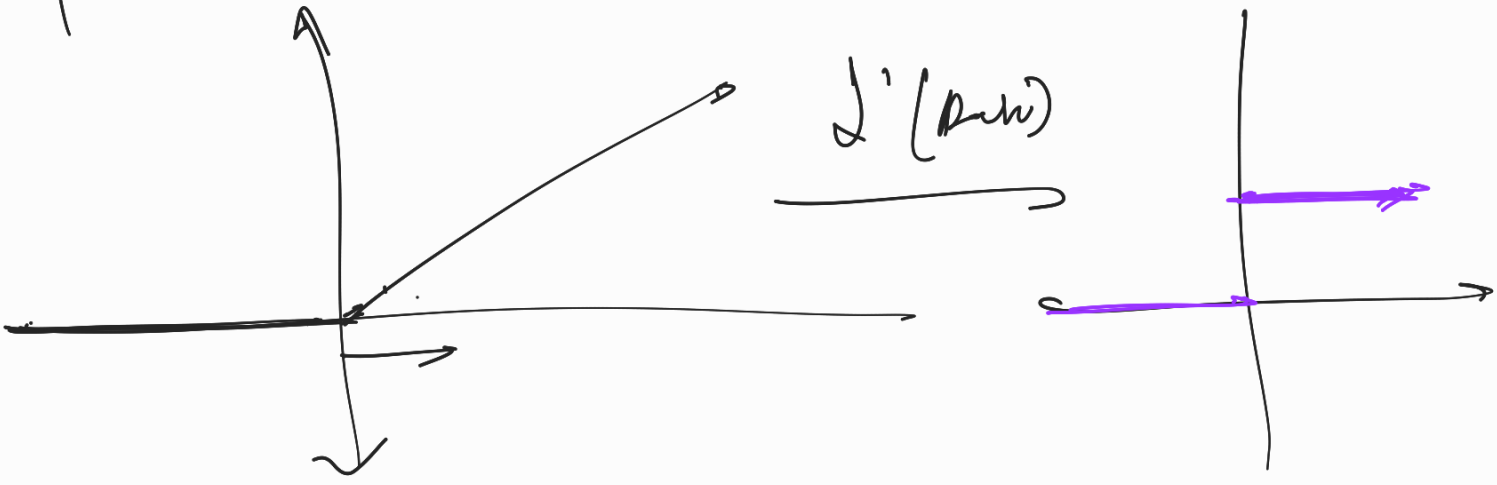
$$w_{i+1} = w_i - \alpha \frac{dh}{dw} =$$

$$\downarrow\downarrow \quad f \left(\begin{matrix} x & x & x & x \\ \uparrow & \uparrow & \uparrow & \uparrow \end{matrix} \right)$$

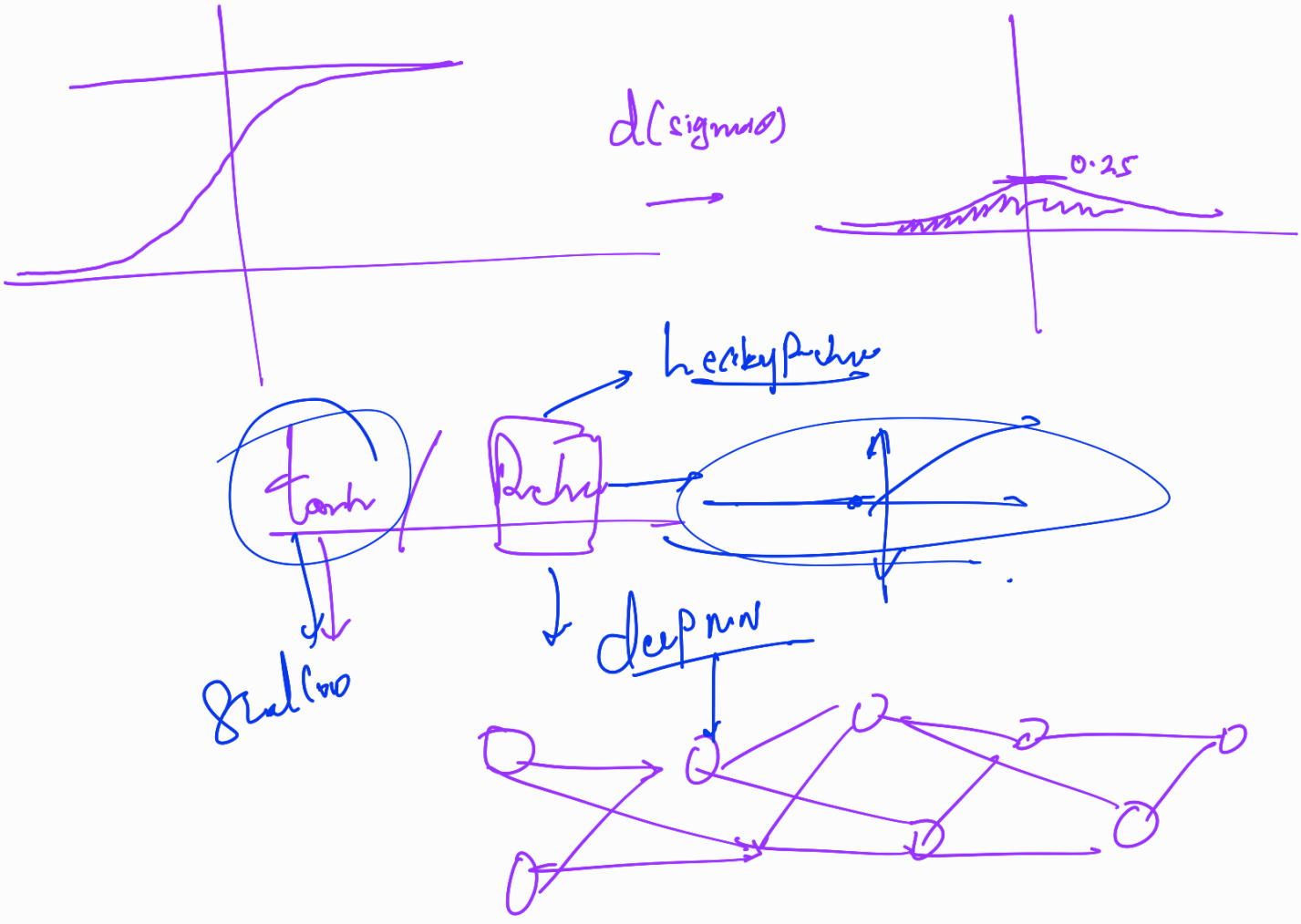


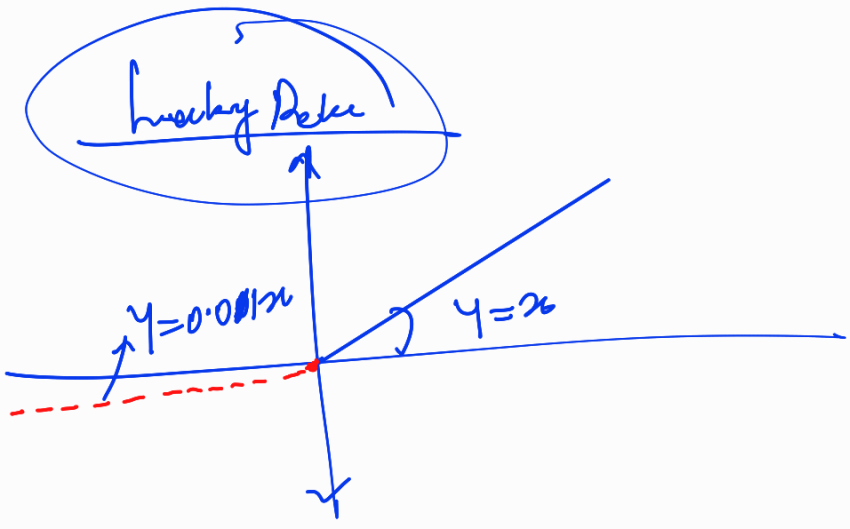
$$w_{i+1} = w_i - \alpha \frac{dh}{dw}$$

Relu

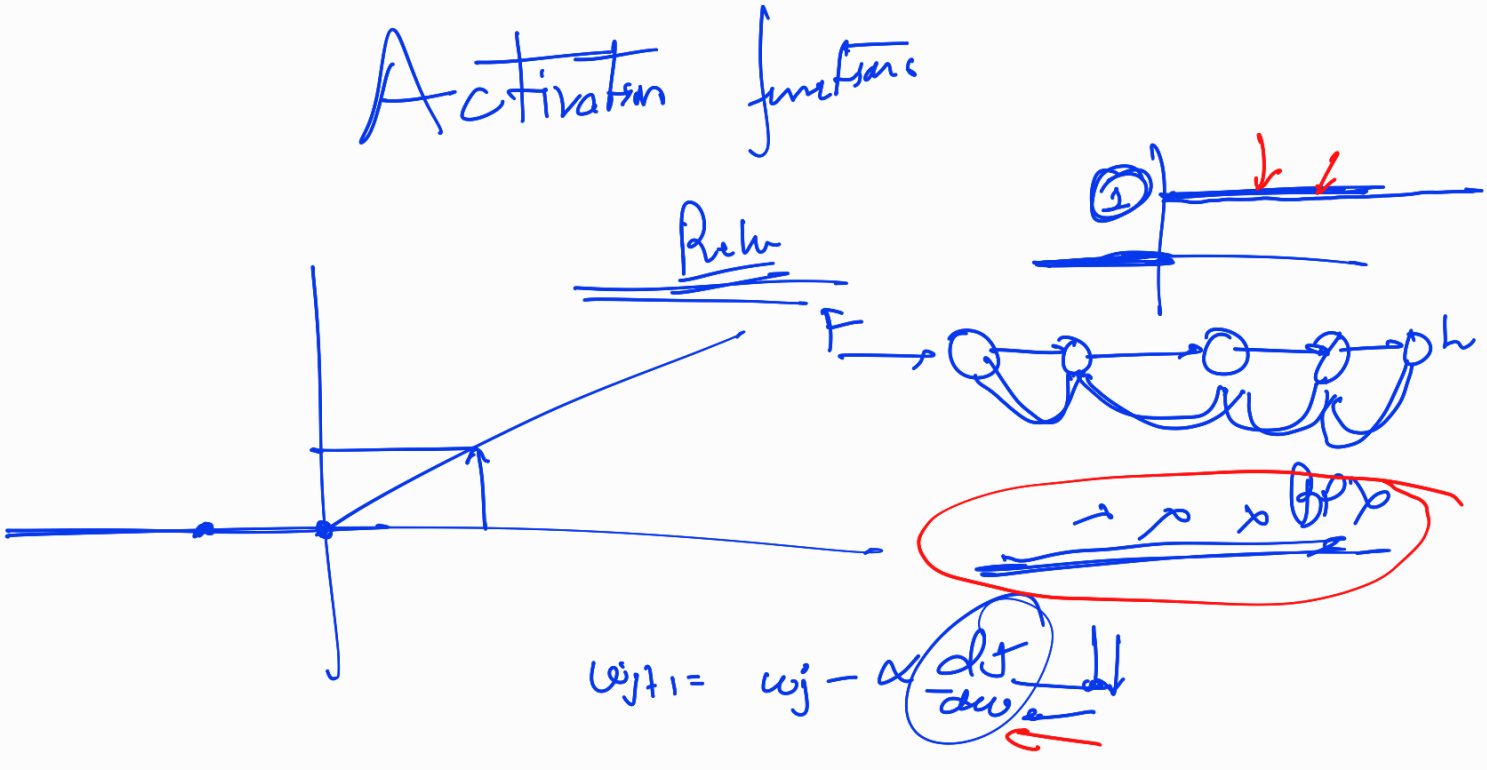


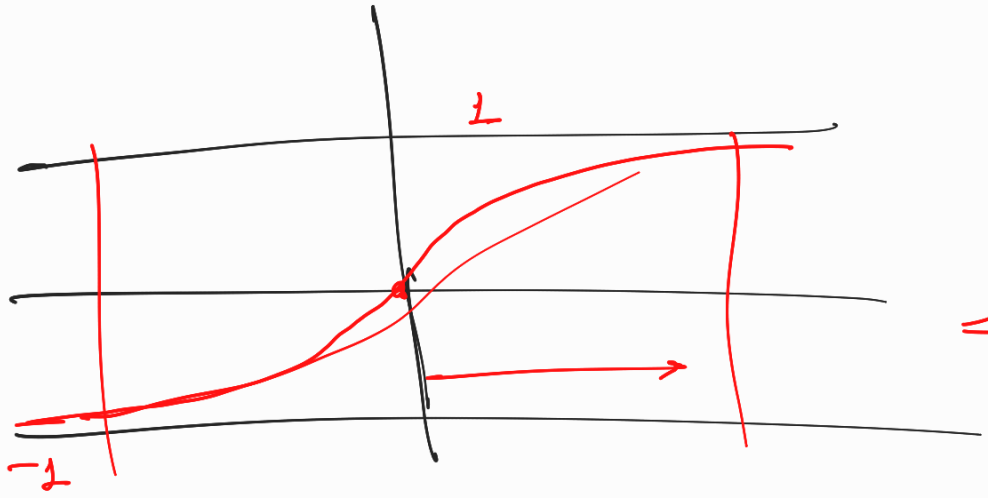
$$w_{i+1} = w_i - \alpha \frac{dh}{dw}$$

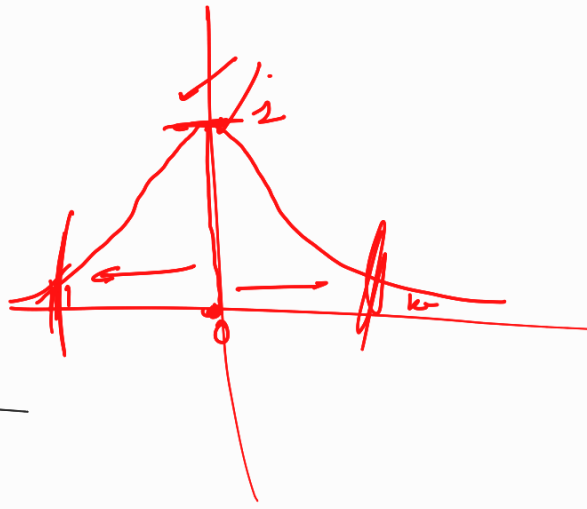

d(sigmoid)
0.25
LeakyReLU
tanh
ReLU
sigmoid
deepnn


Leaky ReLU
y=0.01x
y=x

Activation function







tanh vs Relu

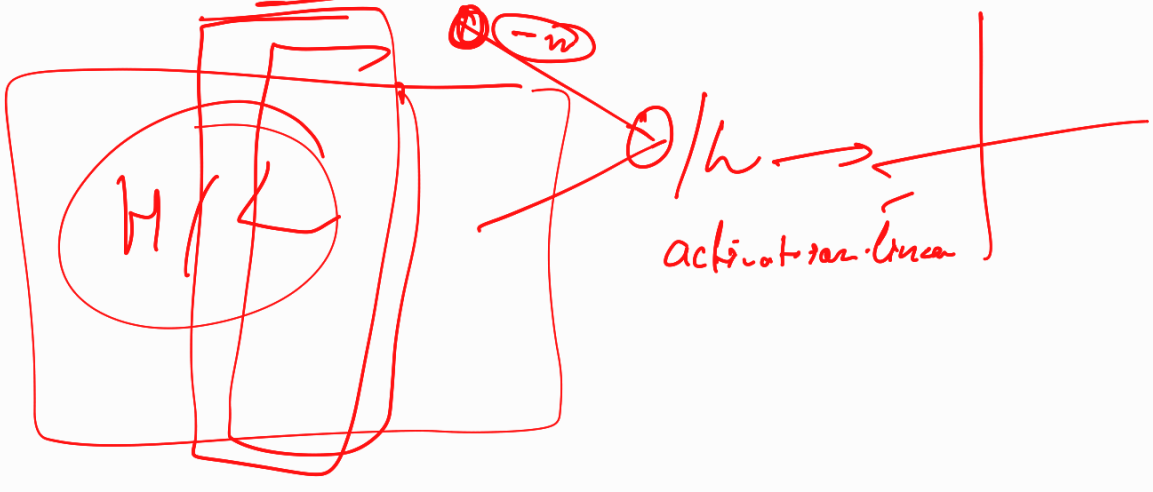
↓ ↓

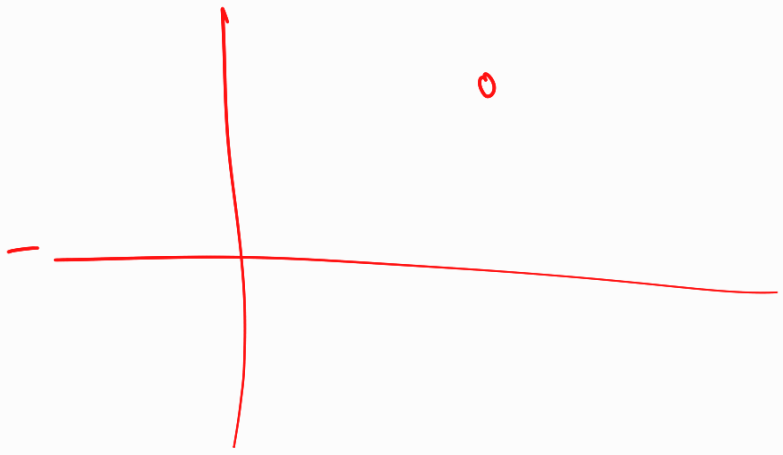
Shallow Deep

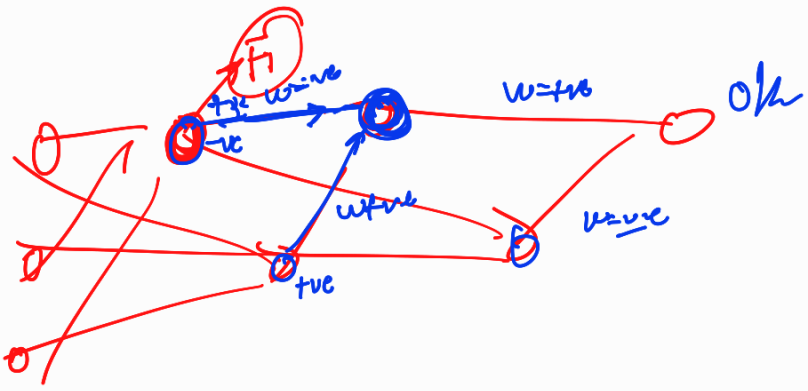
network

x^3 x^5

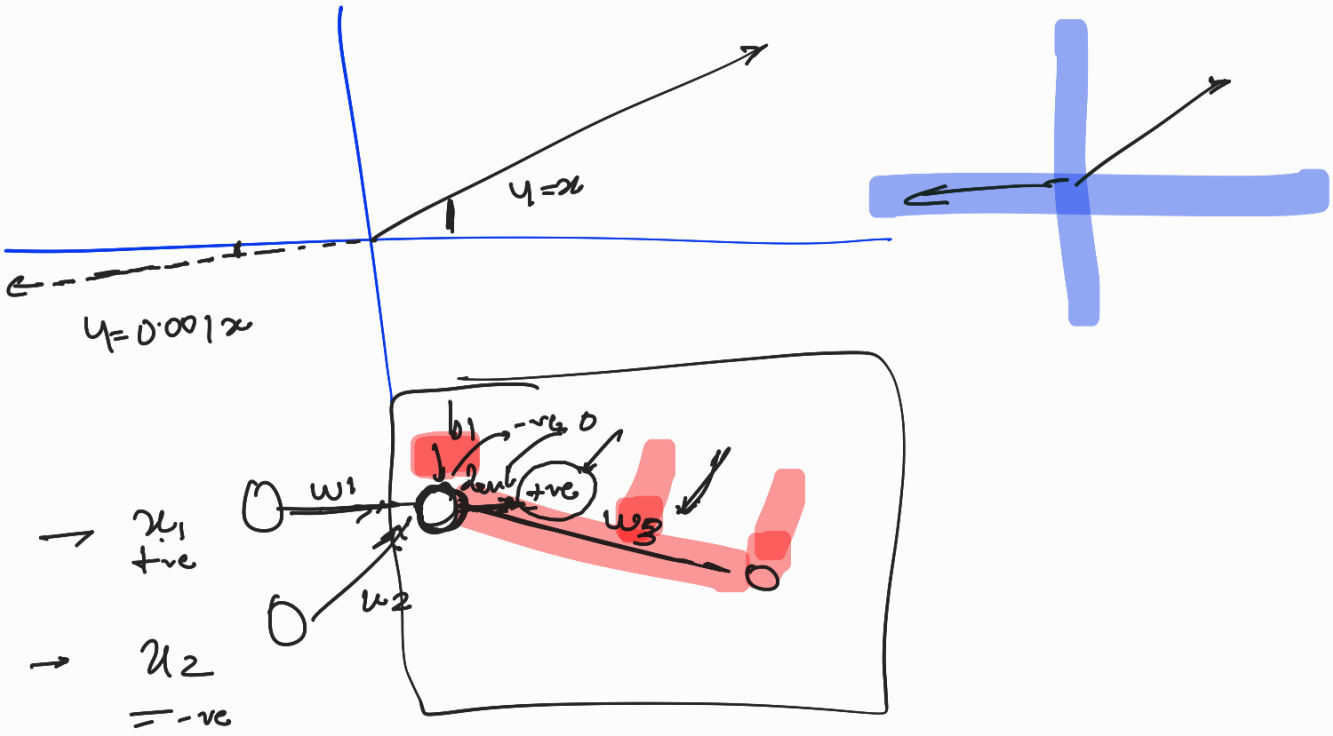
NN (Regression)



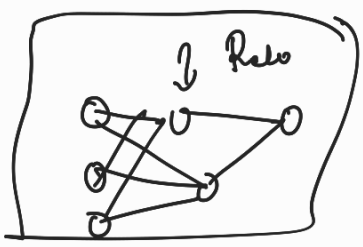

0

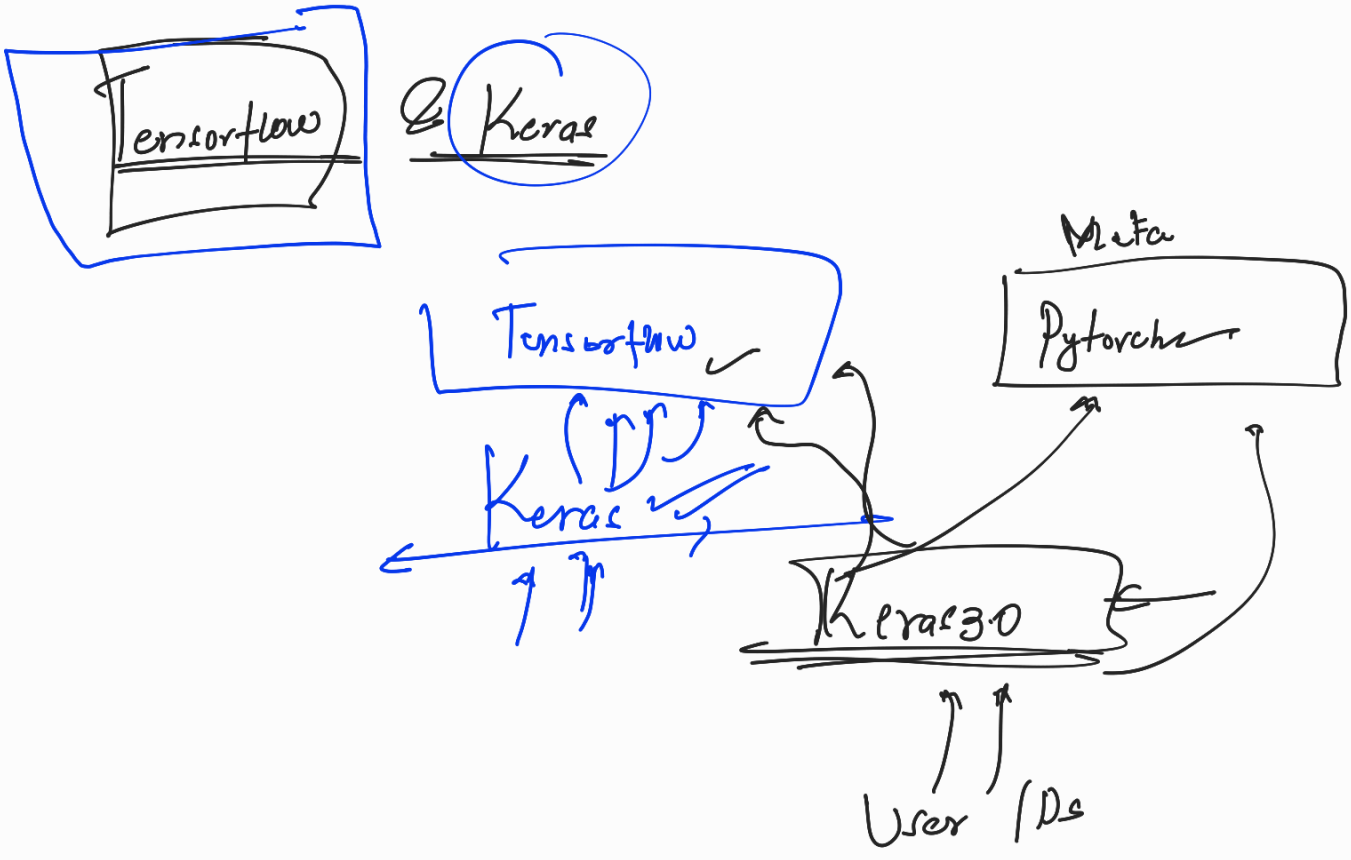

w=+ve
w=+ve
w=+ve
w=-ve
+ve
-ve
+ve
0/p

Leaky ReLu


y=x
y=0.001x
b1
-ve
0
w1
w2
w3
+ve
x1
+ve
x2
=-ve

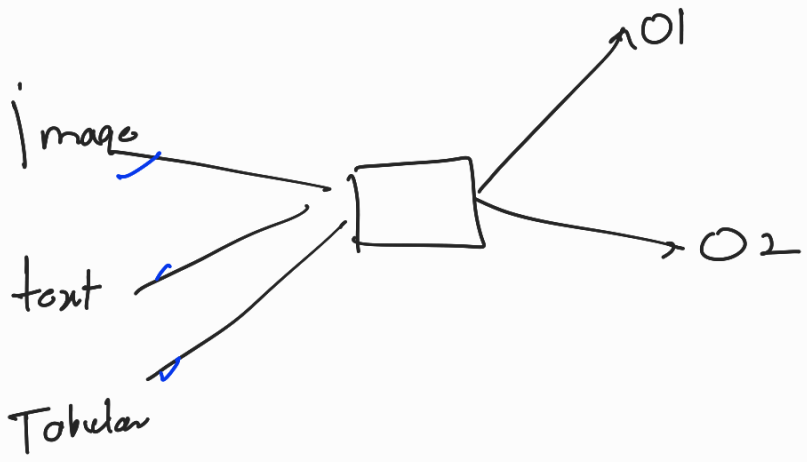
Break !

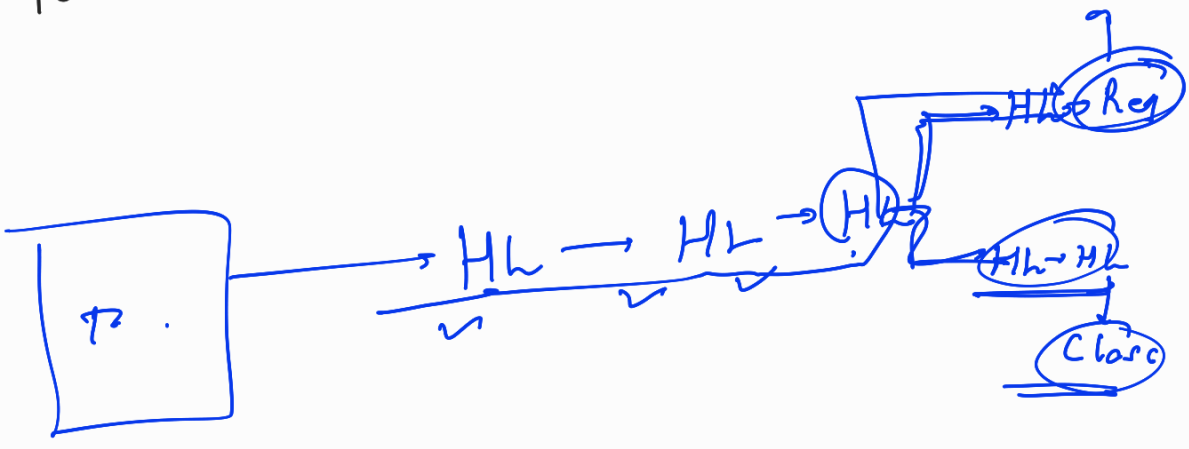




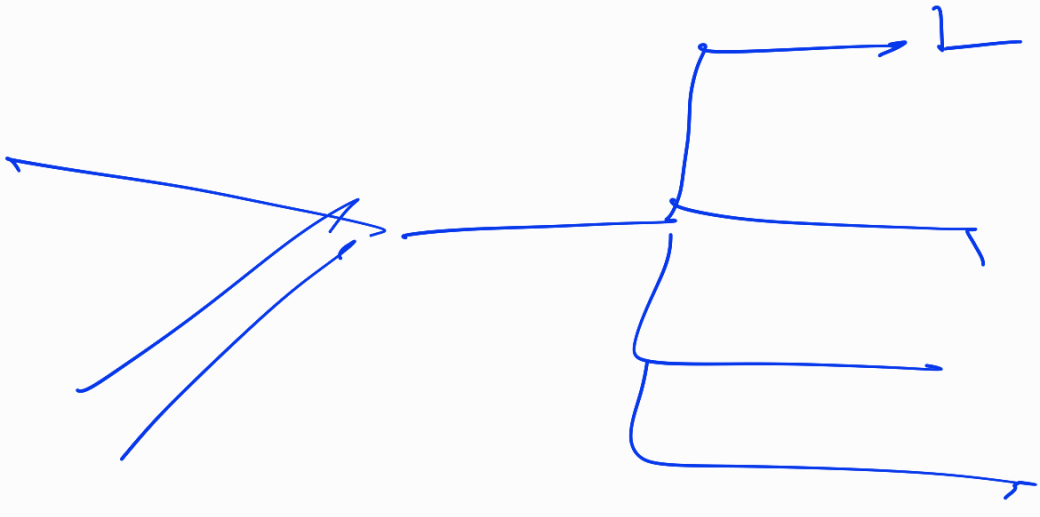
- Simple code on how to implement a nn from scratch in Keras

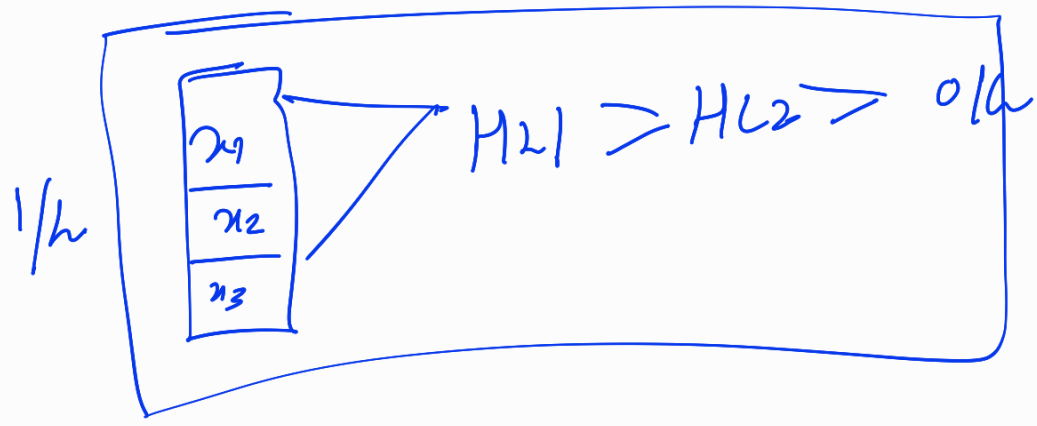
→ improving nn → Dropout, optimizers, Batch Norm., Regularization, Callbacks





I/L → HL → HL → O/L

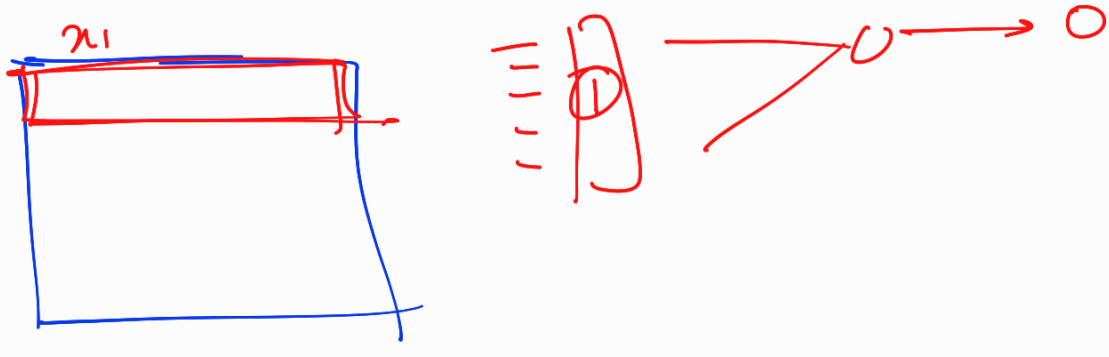




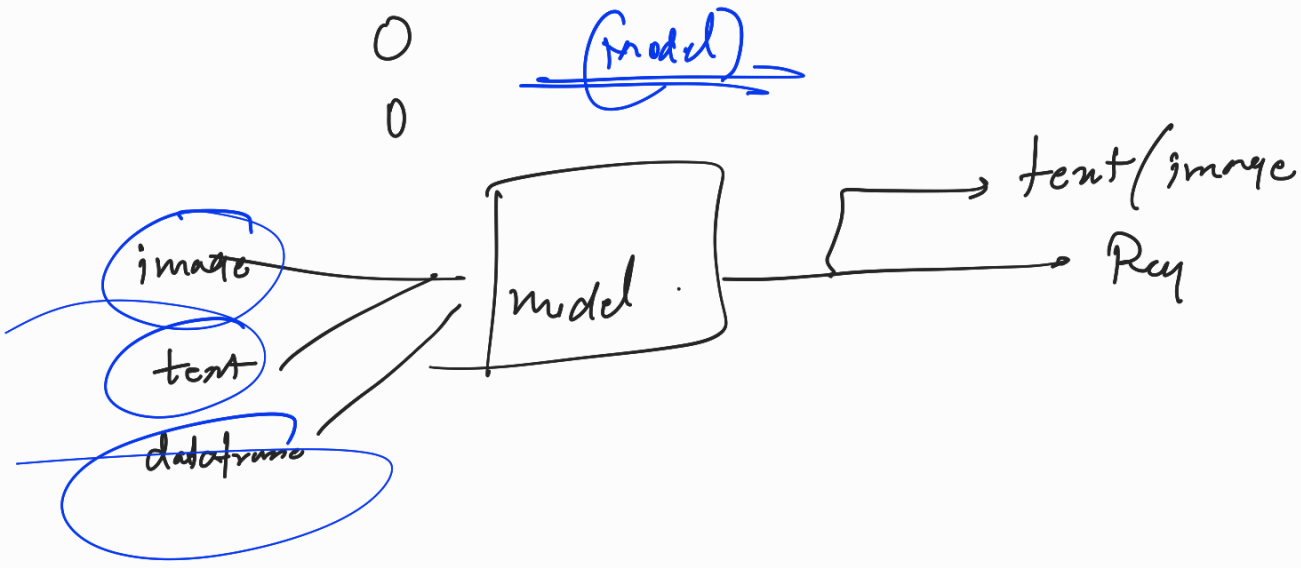
Sequential (1)

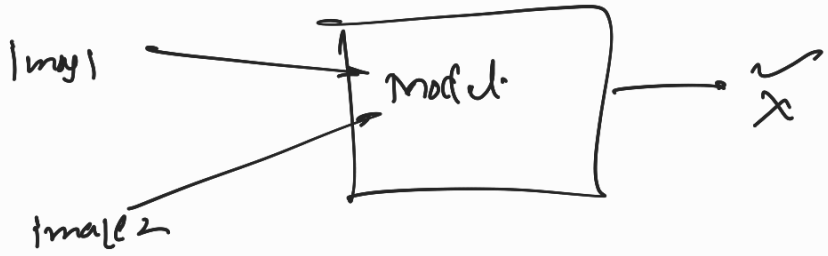
3/05/2024

Sequential



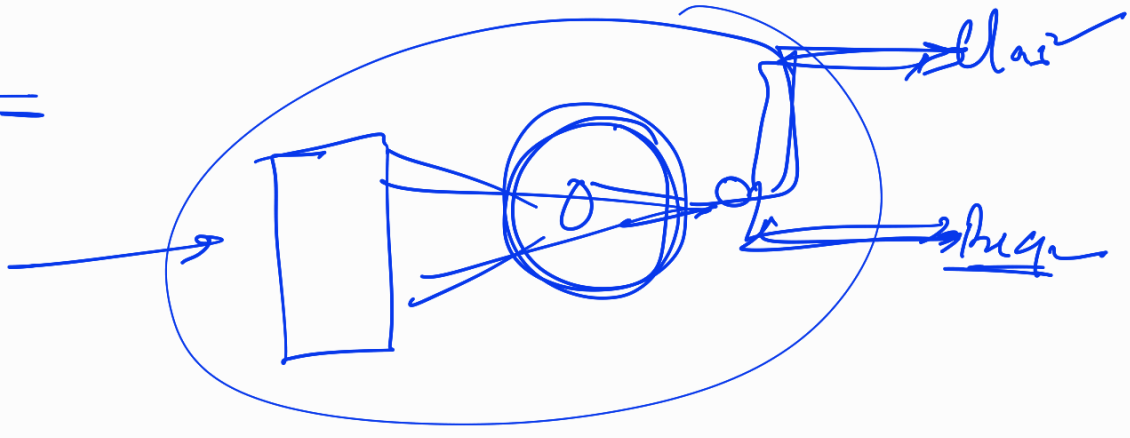
Multimodality:

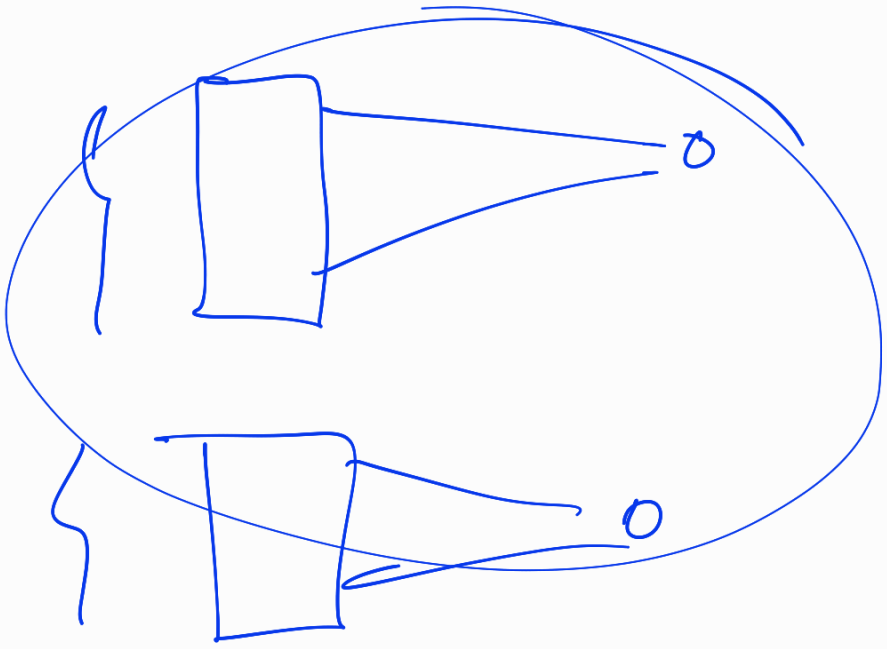


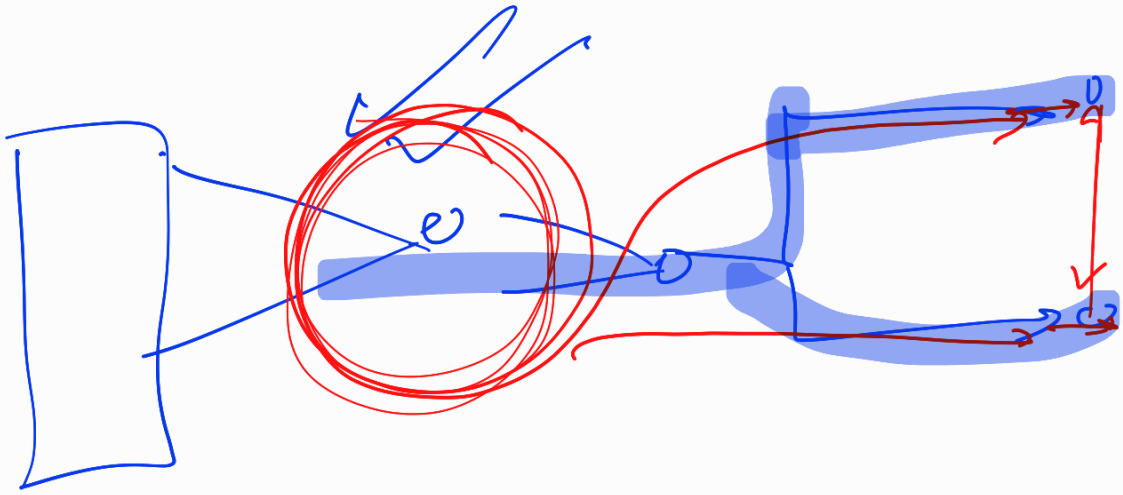


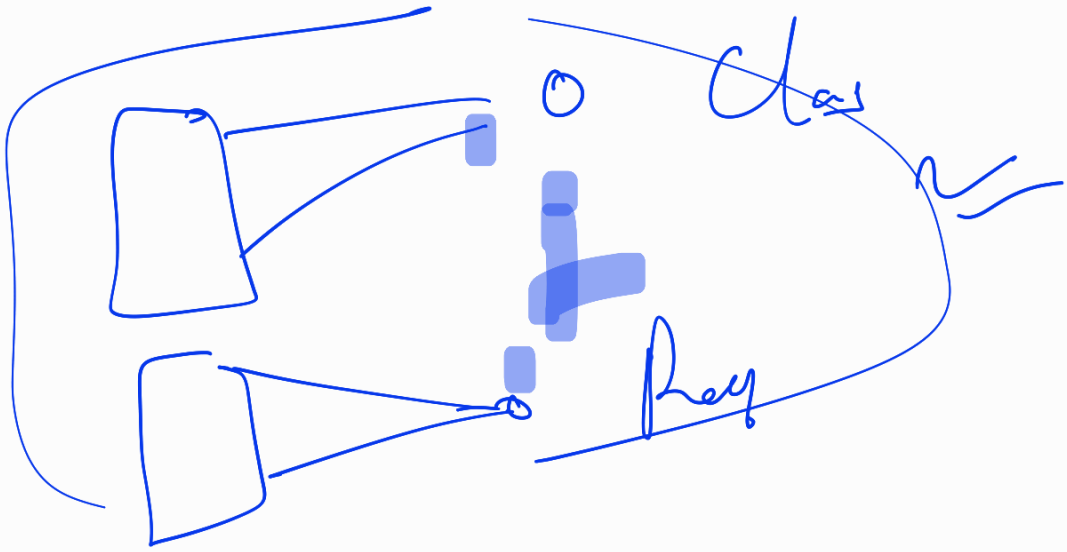
Sequential ()

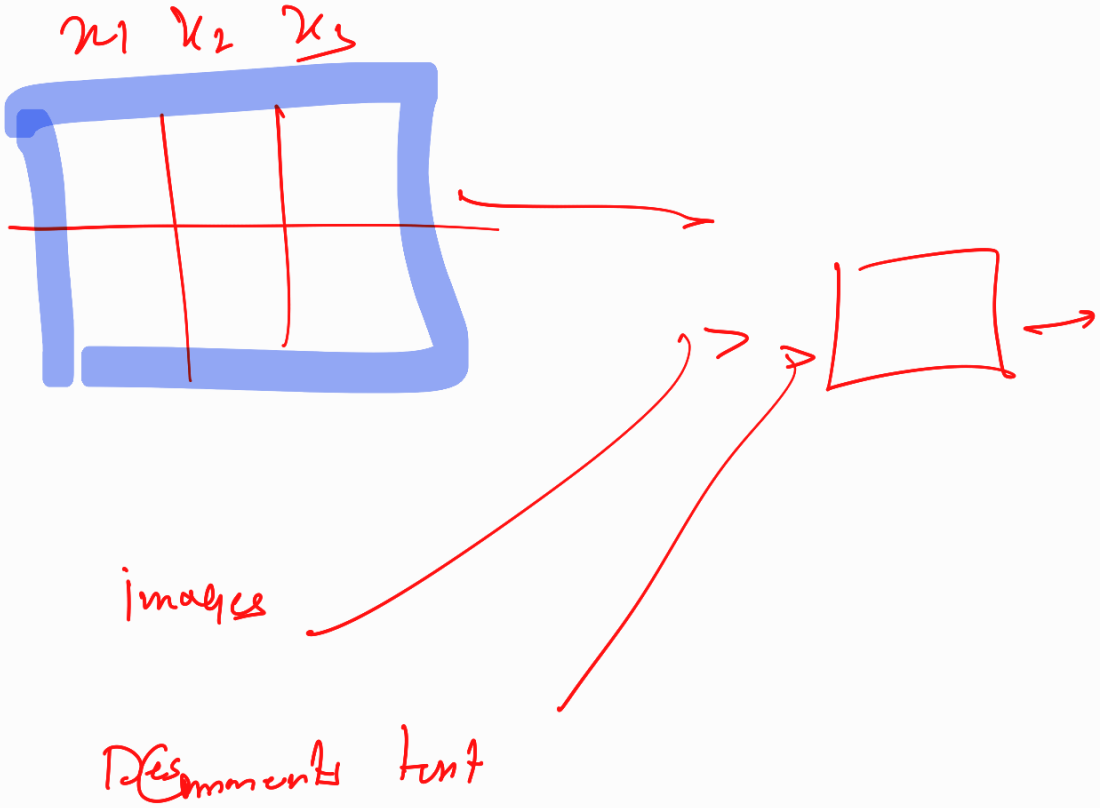
model

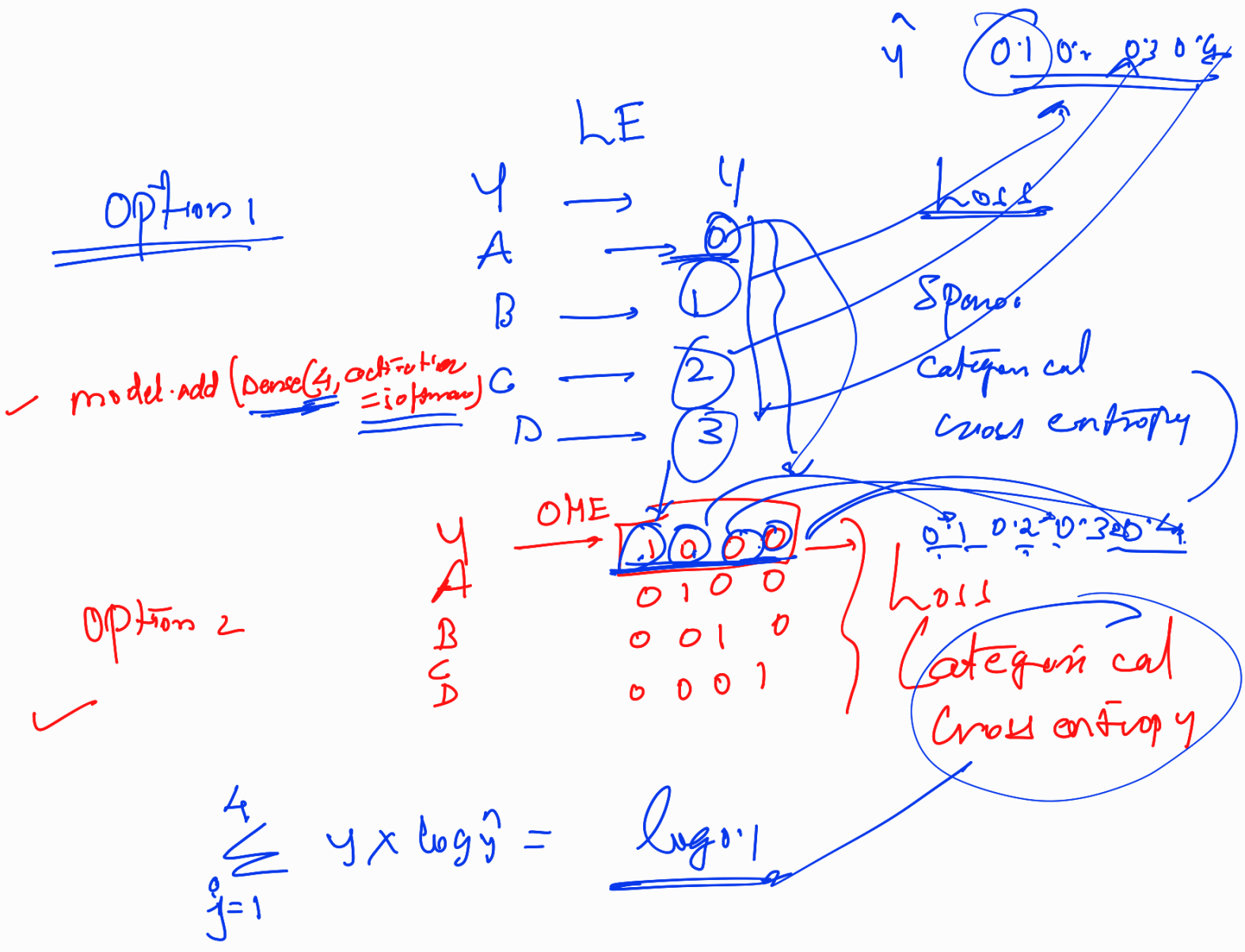


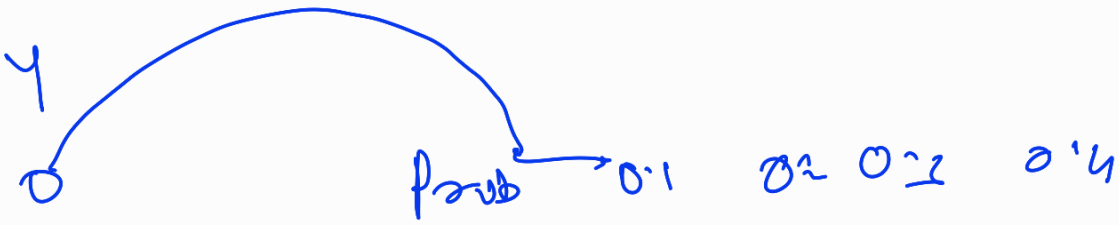






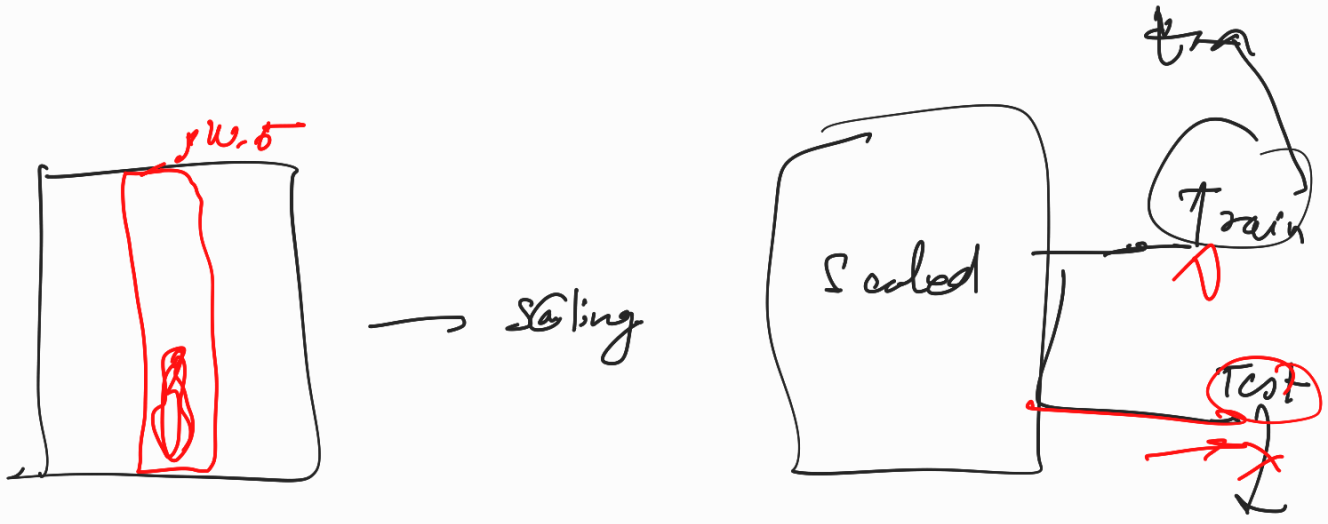

images
Dog
Cat
Horse

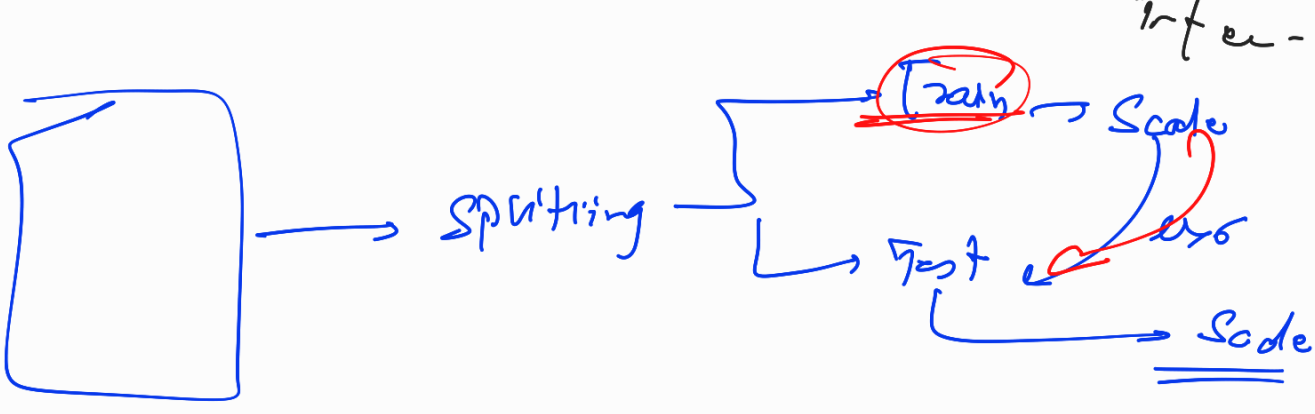

LE
Option 1
model.add(Dense(4, activation='softmax'))
Sparse Categorical cross entropy
Loss
OHE
Option 2
Categorical Cross entropy
$\sum_{j=1}^{4} y \times \log \hat{y} = \log 0.1$

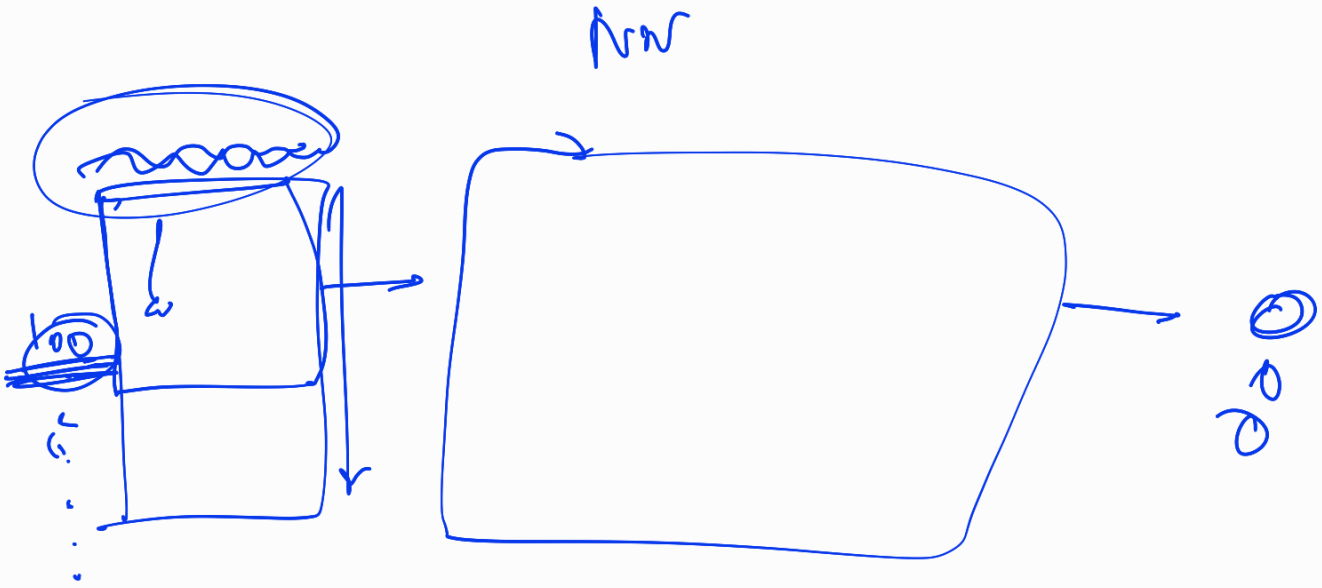


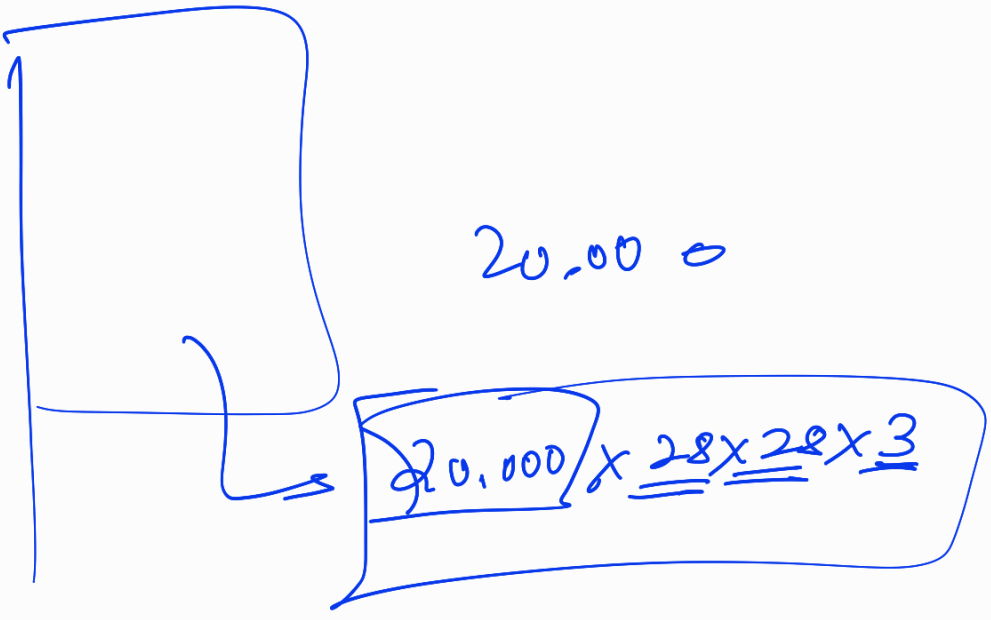
log Probs [0]

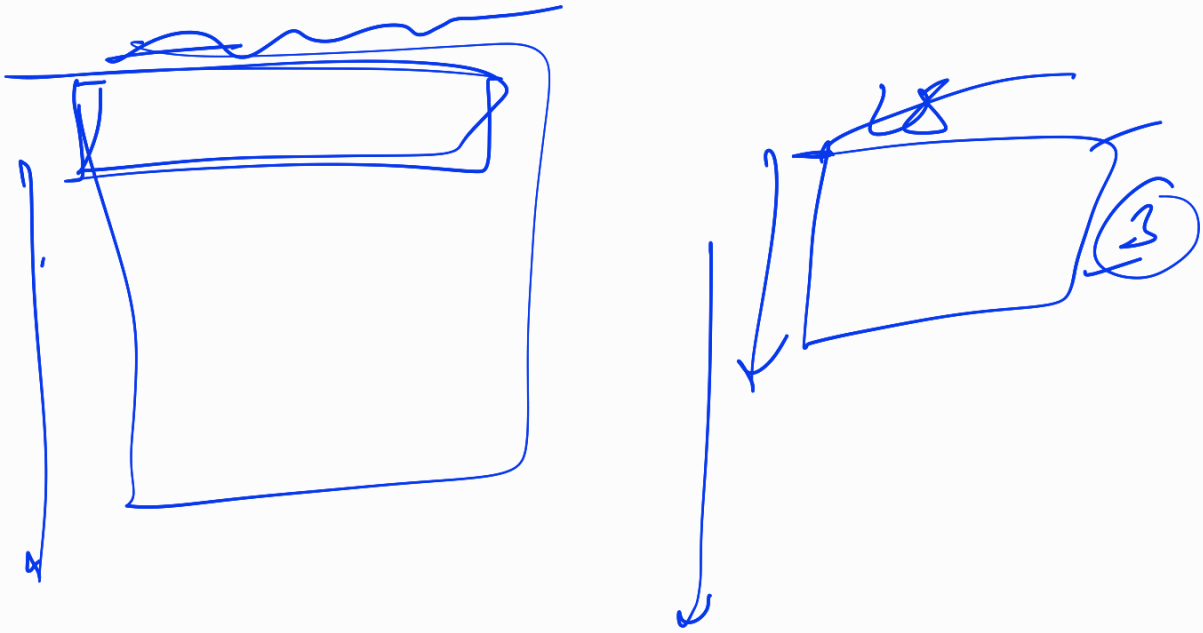
→ 1x log 0.1 + 0x - - - - - - - -

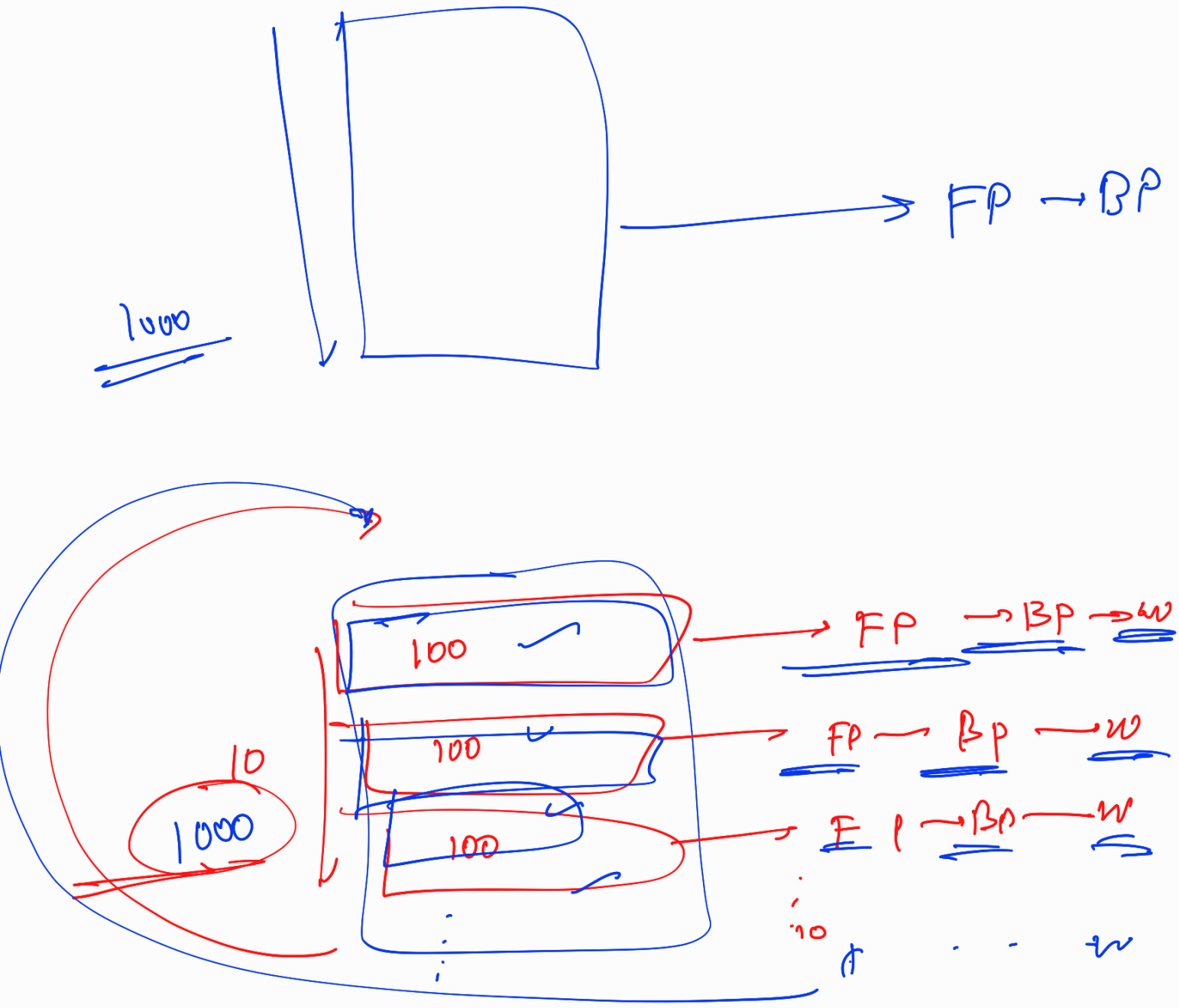


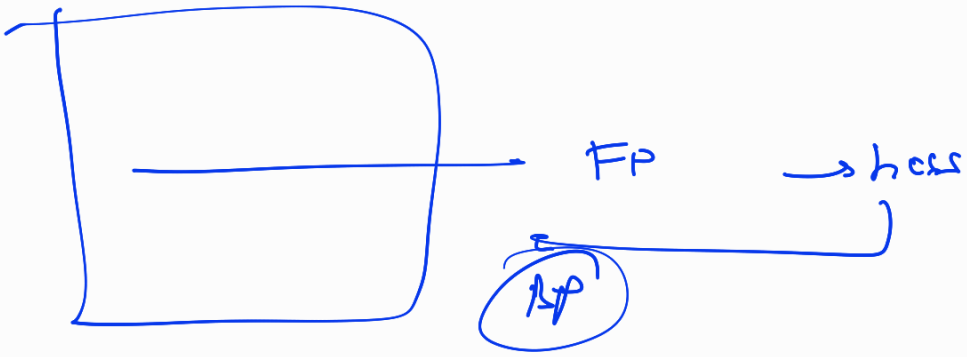


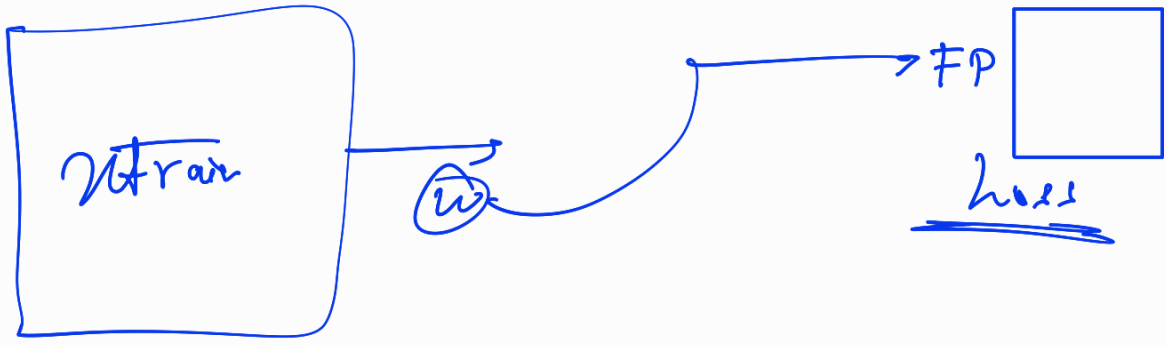


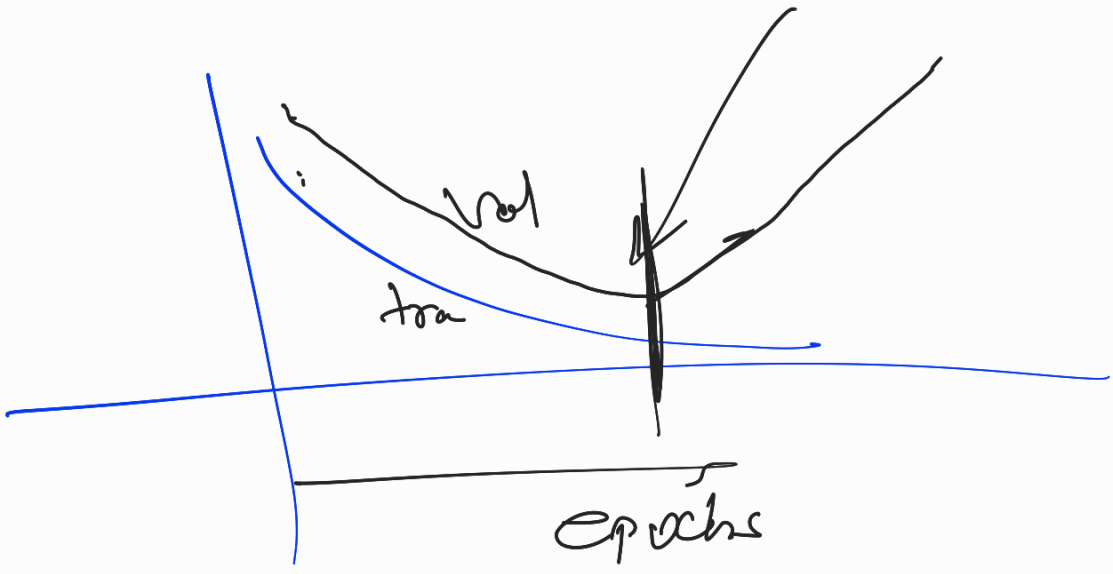

20.00
20,000 x 28x28x3

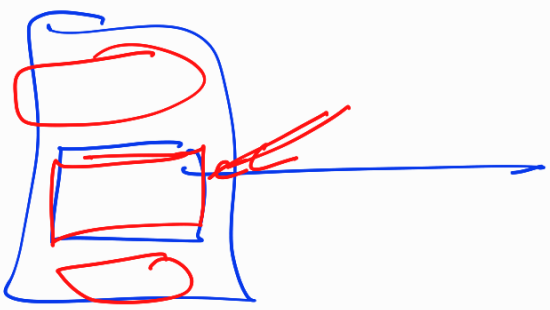

3


1000
FP → BP
10
1000
100
100
100
FP → BP → W
FP → BP → W
FP → BP → W
10





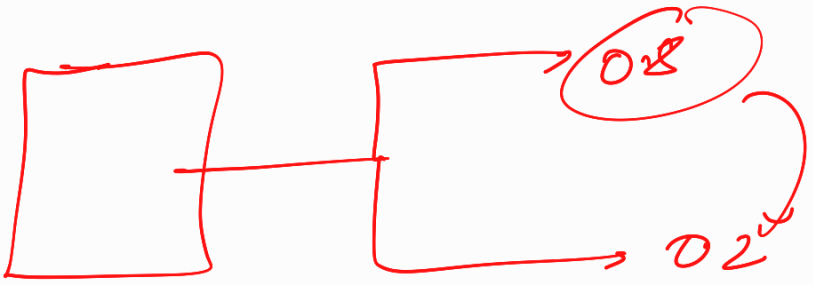


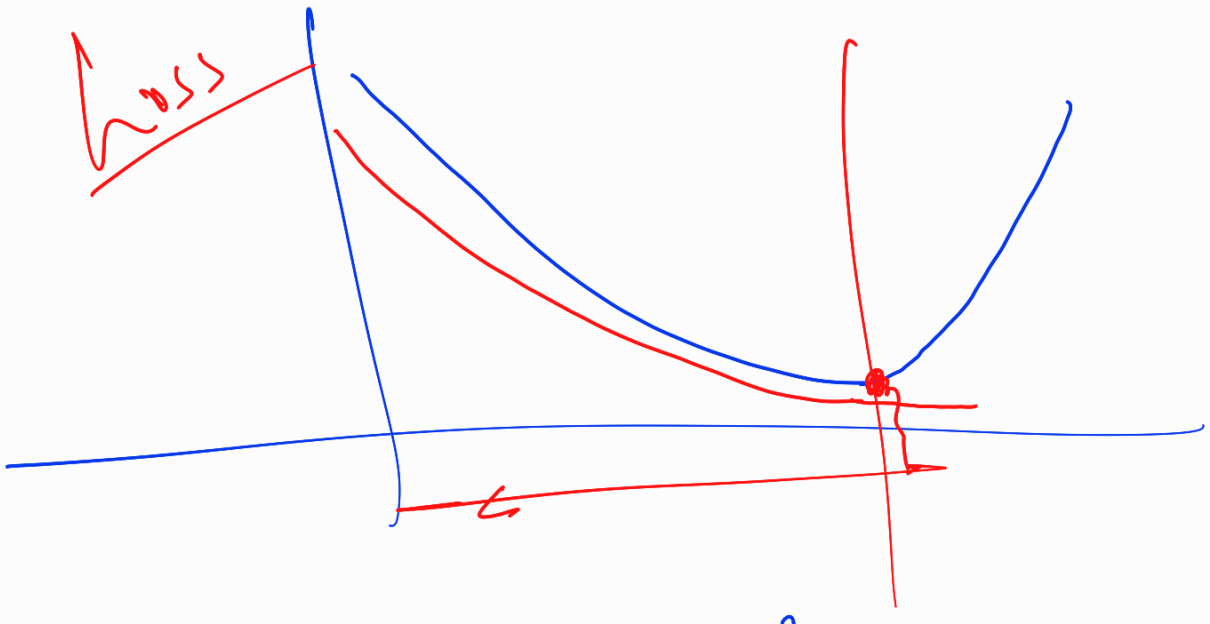


val - 90%

X

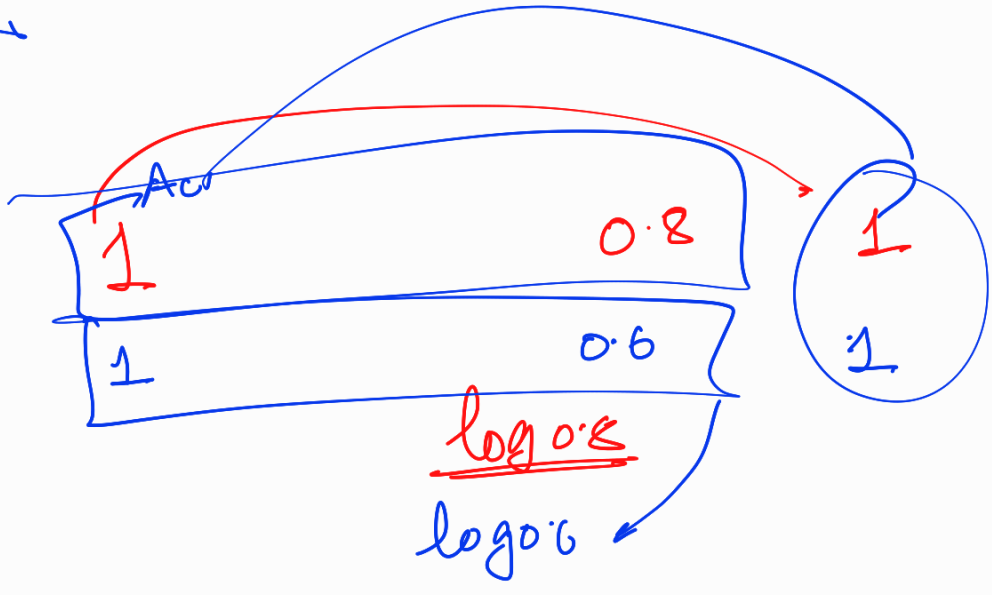
50%

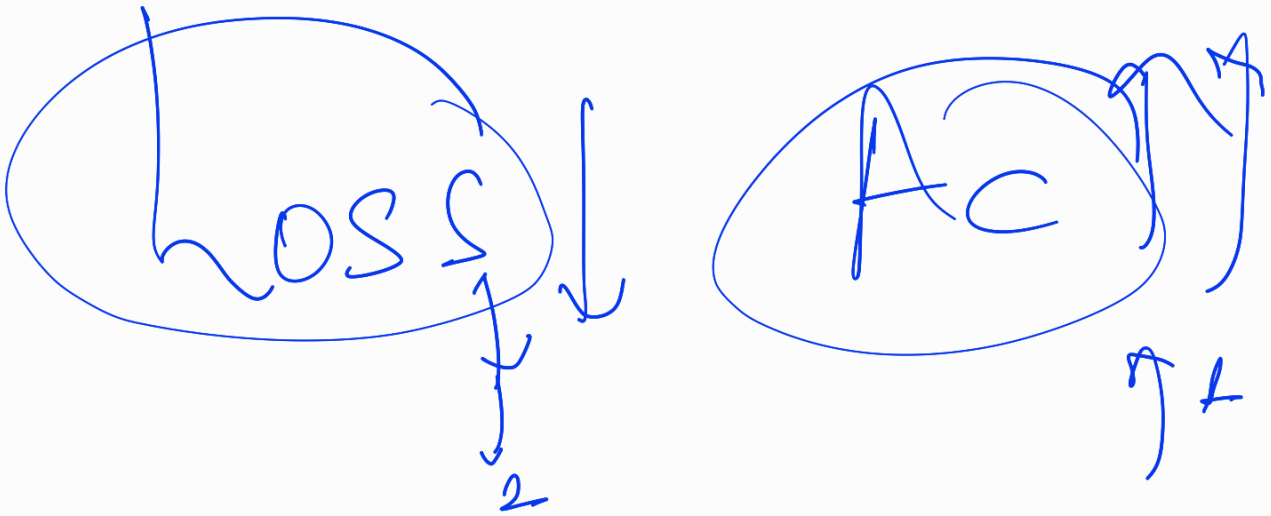




0.32

0.8 ×





Loss → log loss

1 0 0 0

0.35 0.31 0.3 0.04 → 1

0.8

log 0.35

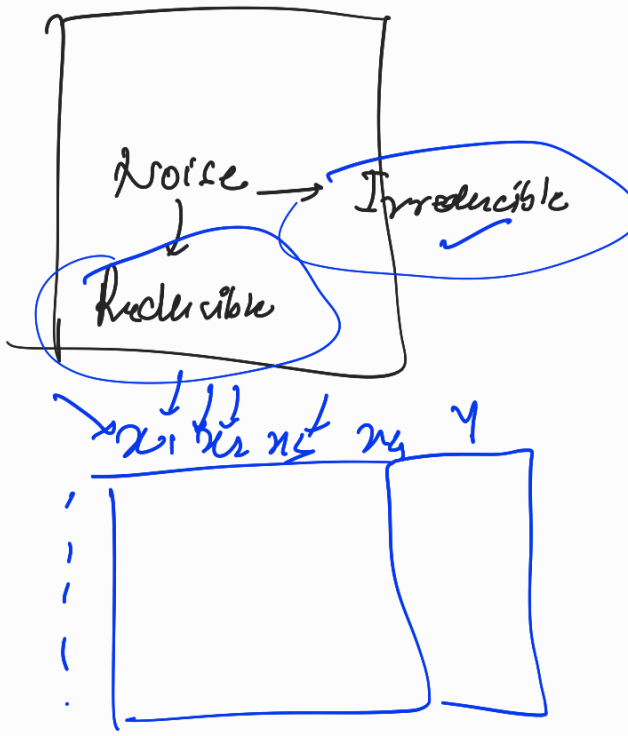
100%

100

Trained → loss Accuracy

6/5/2024

- Dropout
- ① Regularization
- ② optimizers
- ③ Batch normalization
- ④ Hyperparameter tuning
- ⑤ Callbacks
- ⑥ weight initializers

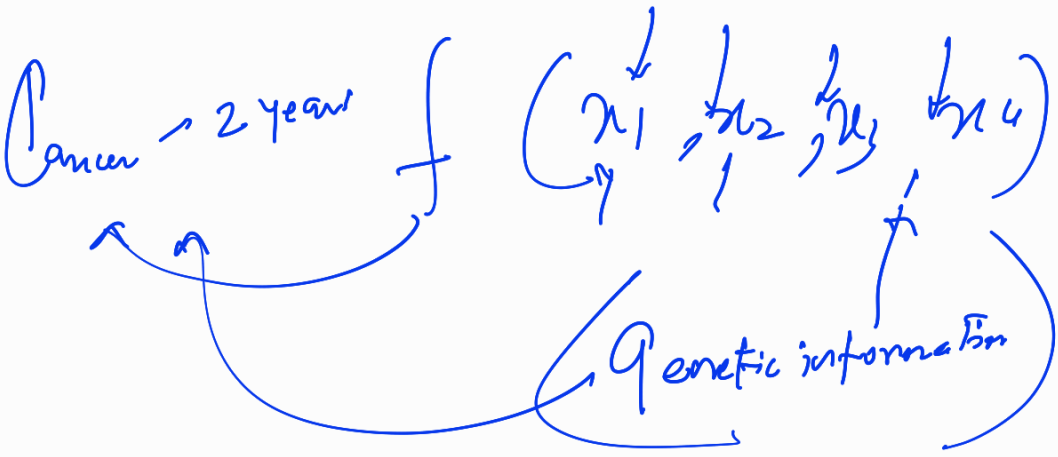


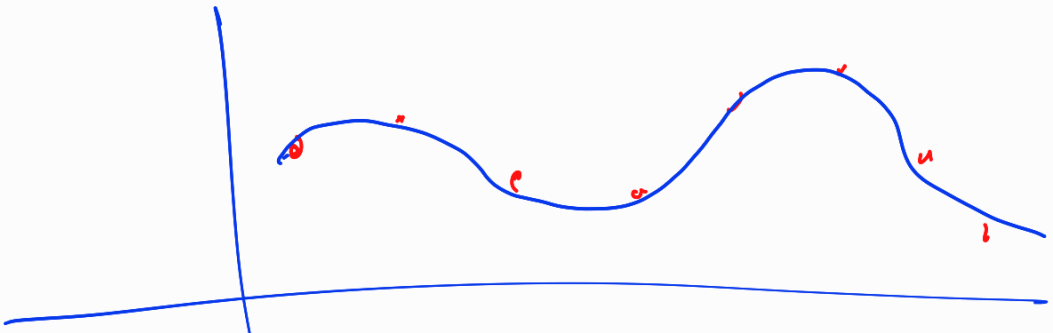
$y \sim f(x_1, x_2, x_3, x_4)$

DT, NN, LR.

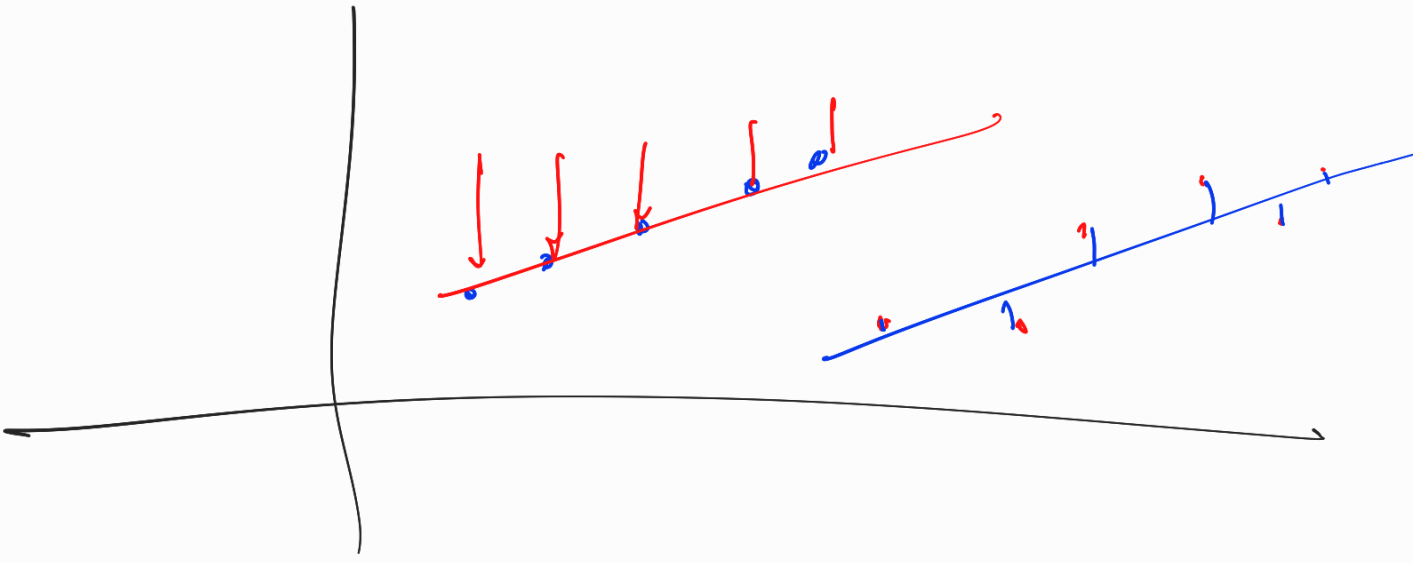
$y_n \leftarrow f(x_1, x_2, x_3, x_4)$

x_7, x_6

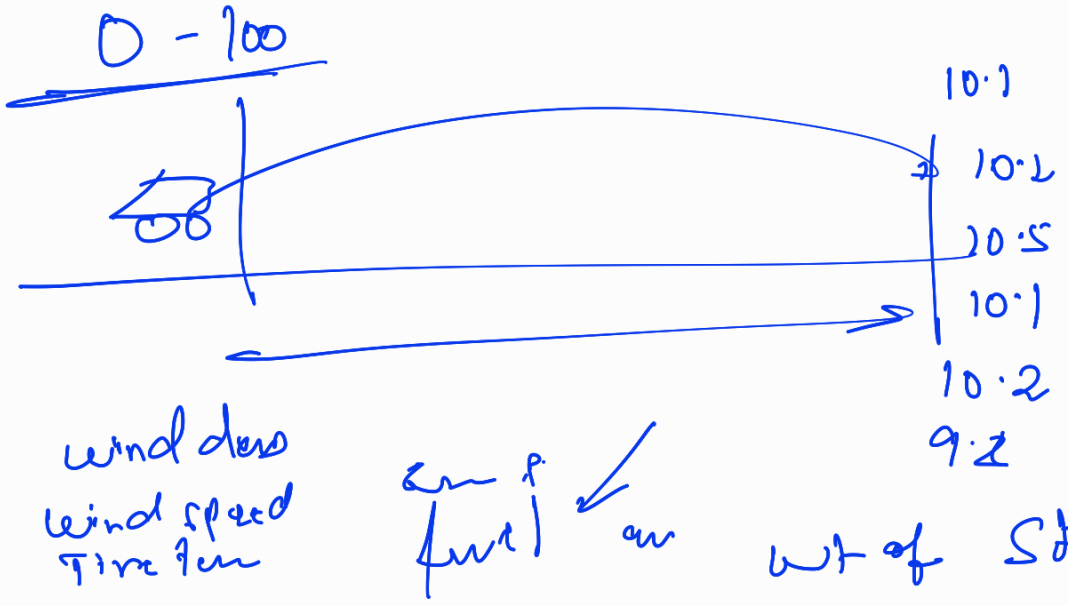


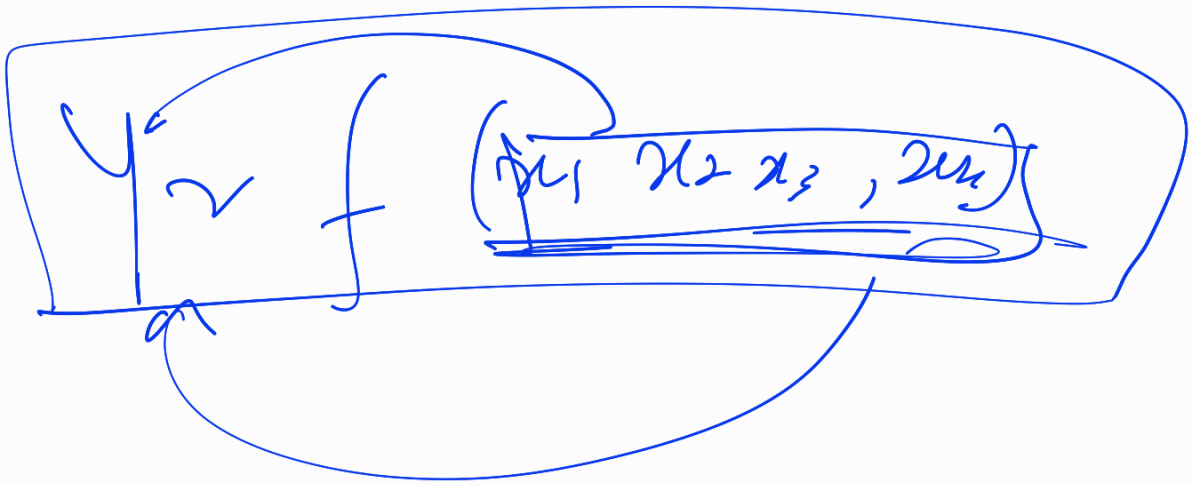


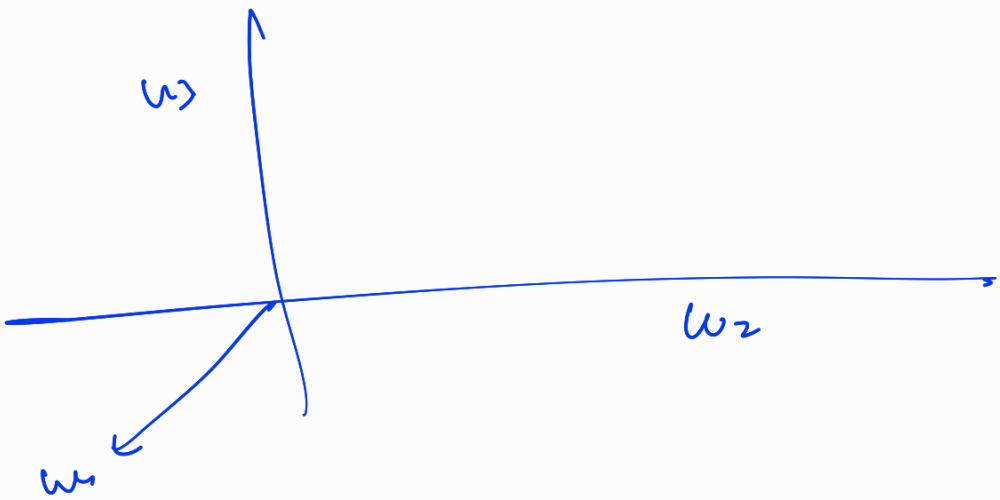
Bias Variance

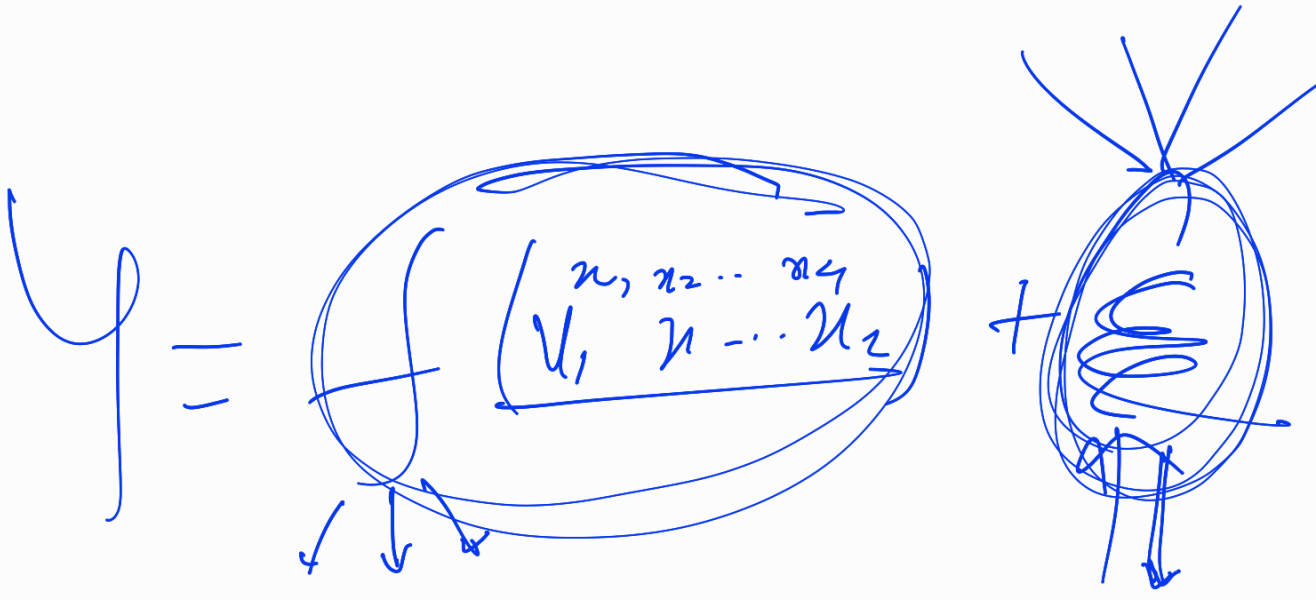



Non-convex
Convex optimization


0 - 100
10.1
10.2
10.5
10.1
10.2
9.2
wind does
wind speed
Tire pen
wt of Stop watch




w_3
w_2
w_1



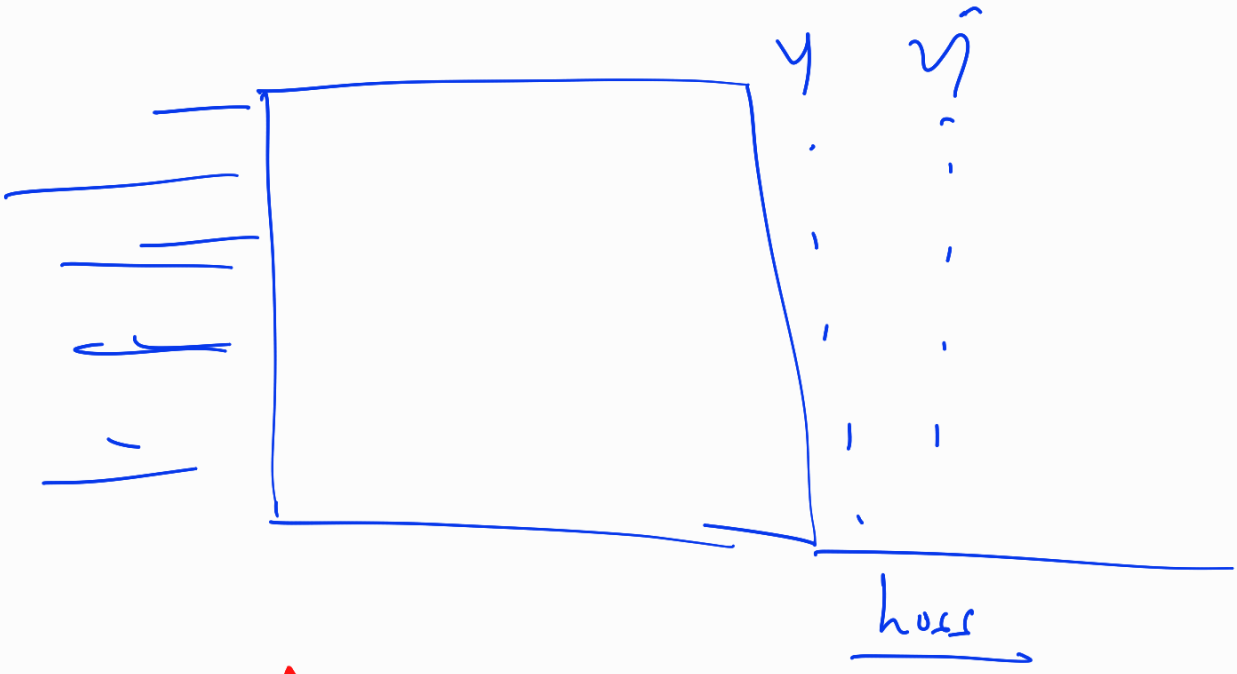
x_0, x^2

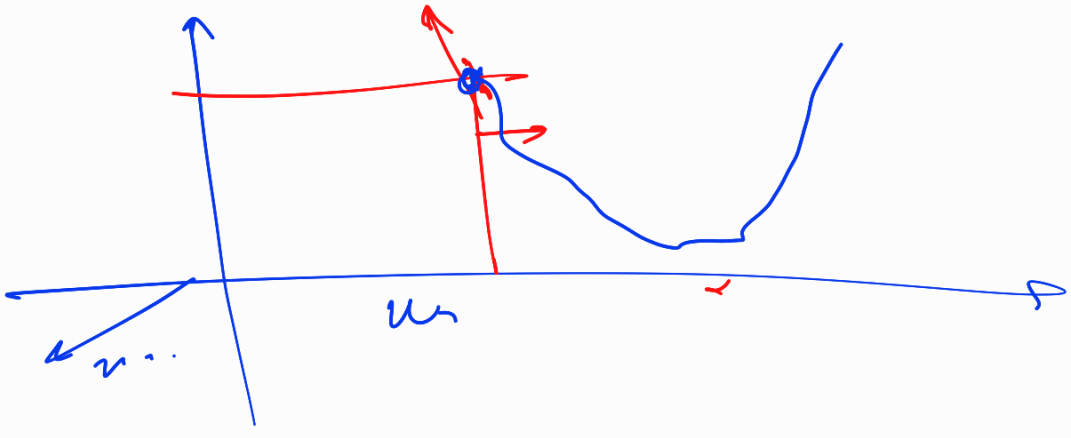
x_1 → x_1^2

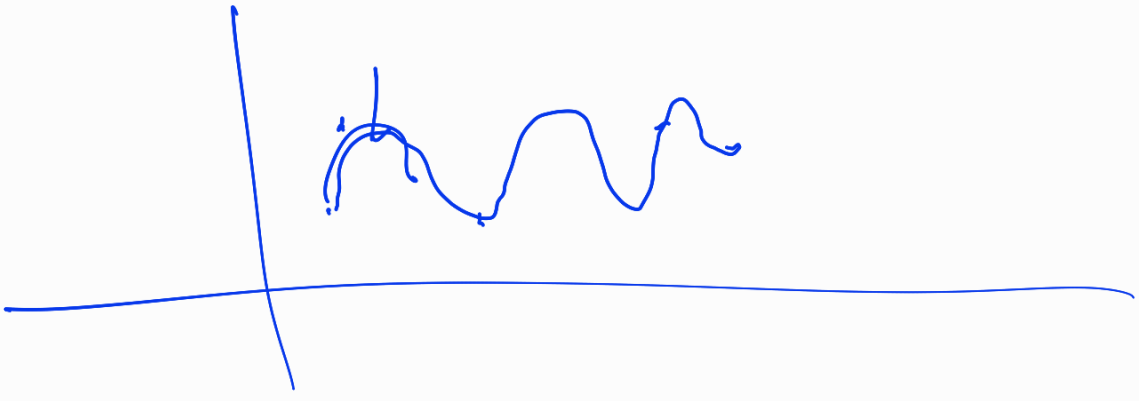
x_5

Temp ↓

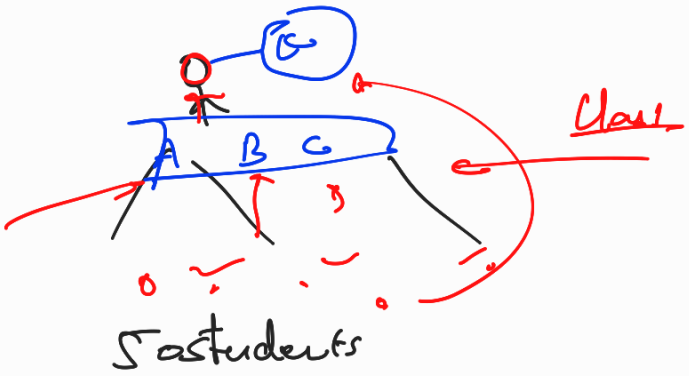
Wind


y
ŷ
loss

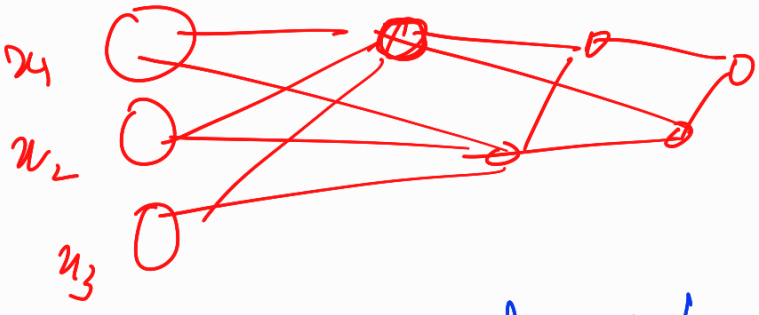




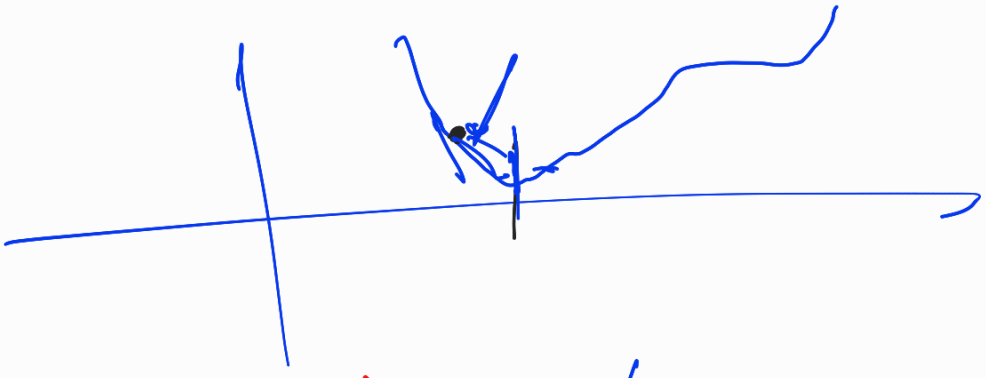
Drop out:

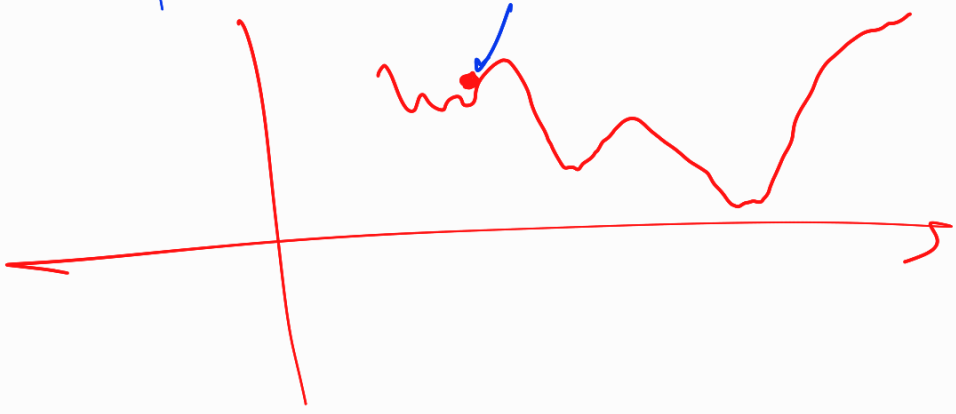


Cold calling

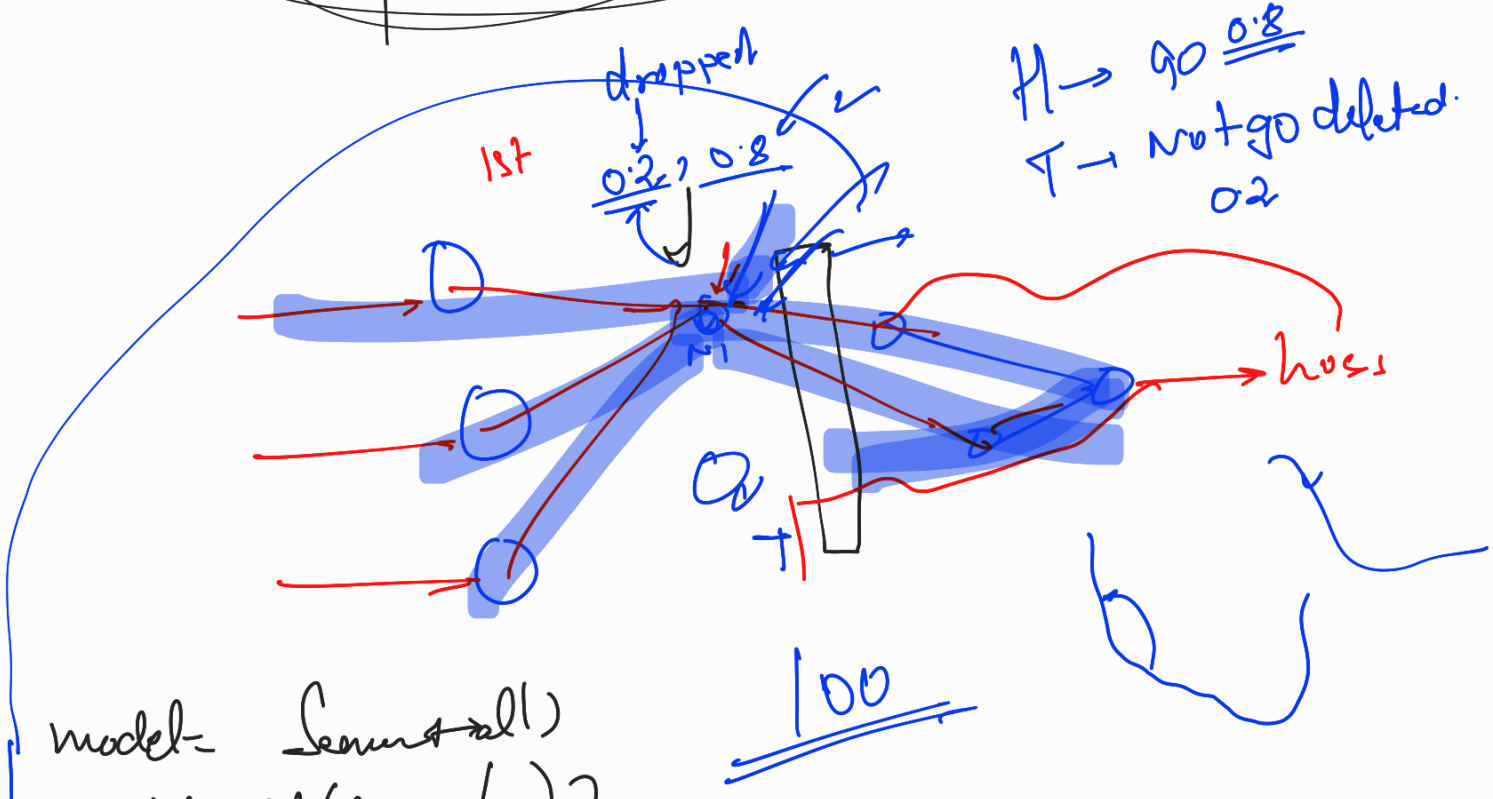


is every answer not equal for me





Optimizers

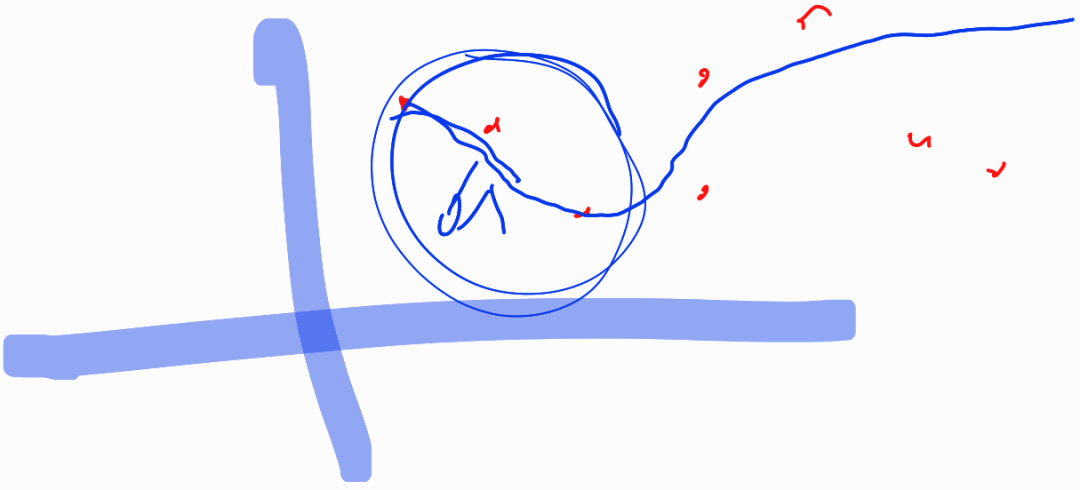


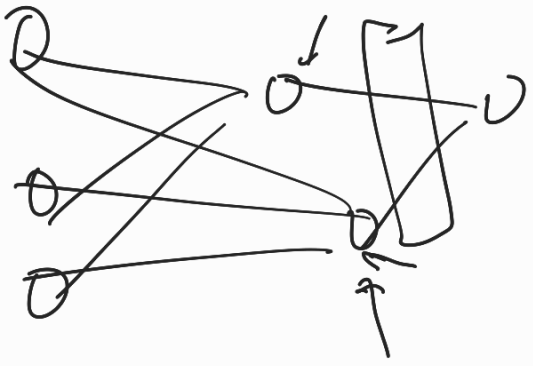
model = Sequential()

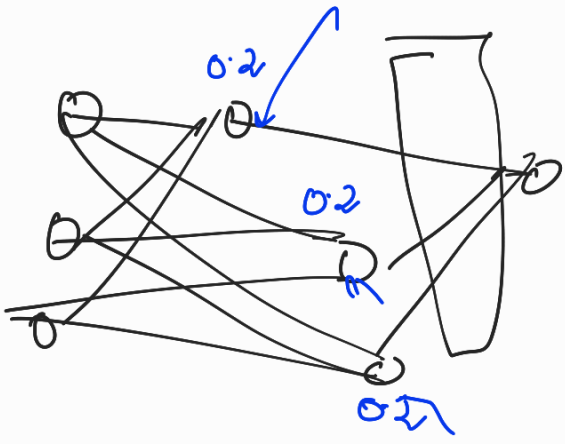
model.add(Dense(2))

model.add(Dropout(dropout_rate)) 0.2

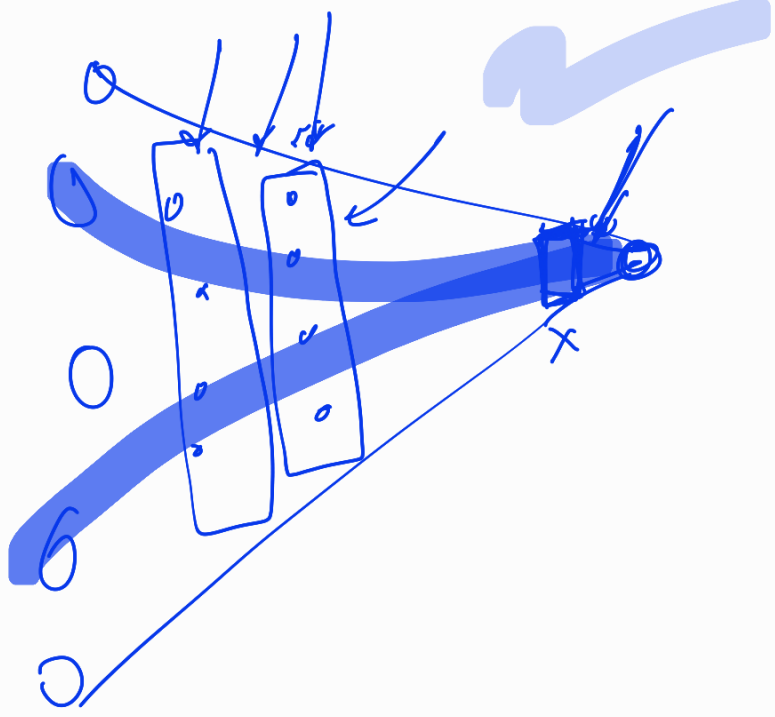
↑ Hyp



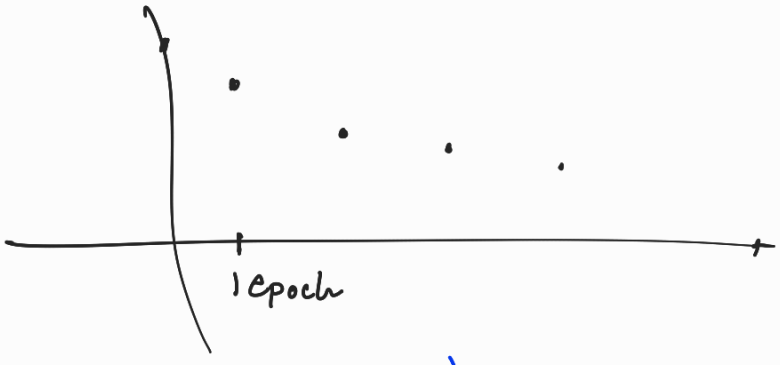


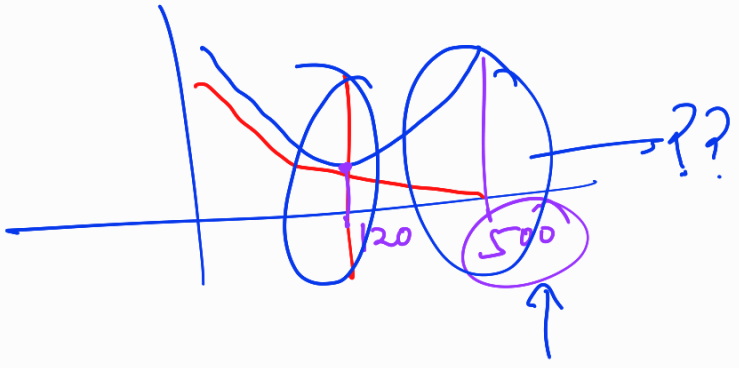


$${}^{3}C_{2}\,0.2\times0.2$$



Callbacks

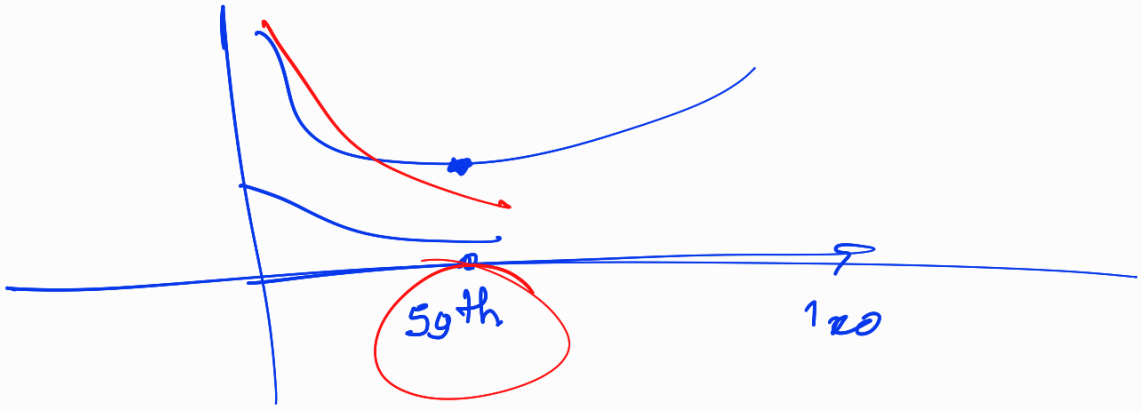




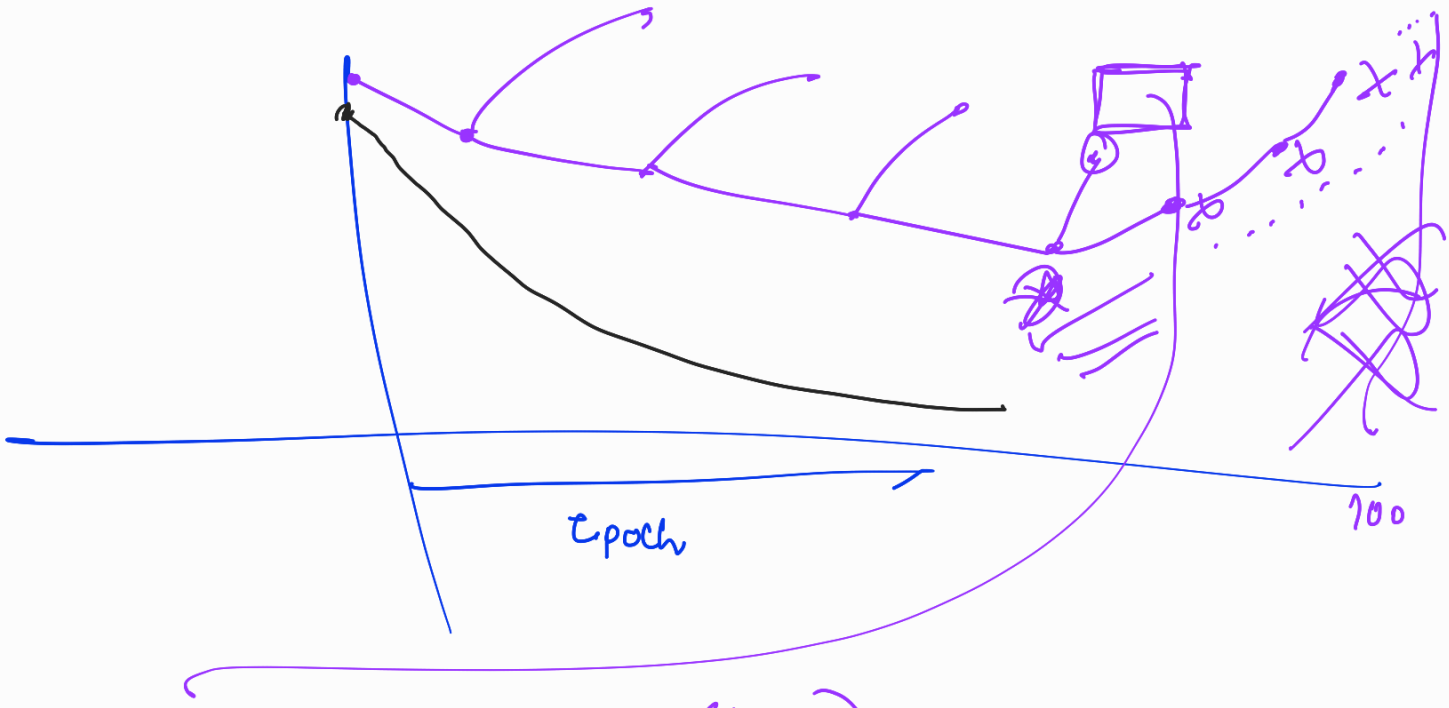
Epochs=500

Model Checkpoint →

→ after every epoch and if my val-loss has reduced from the previous epoch's val-loss of the model I will save the weights at a particular file path else will continue.

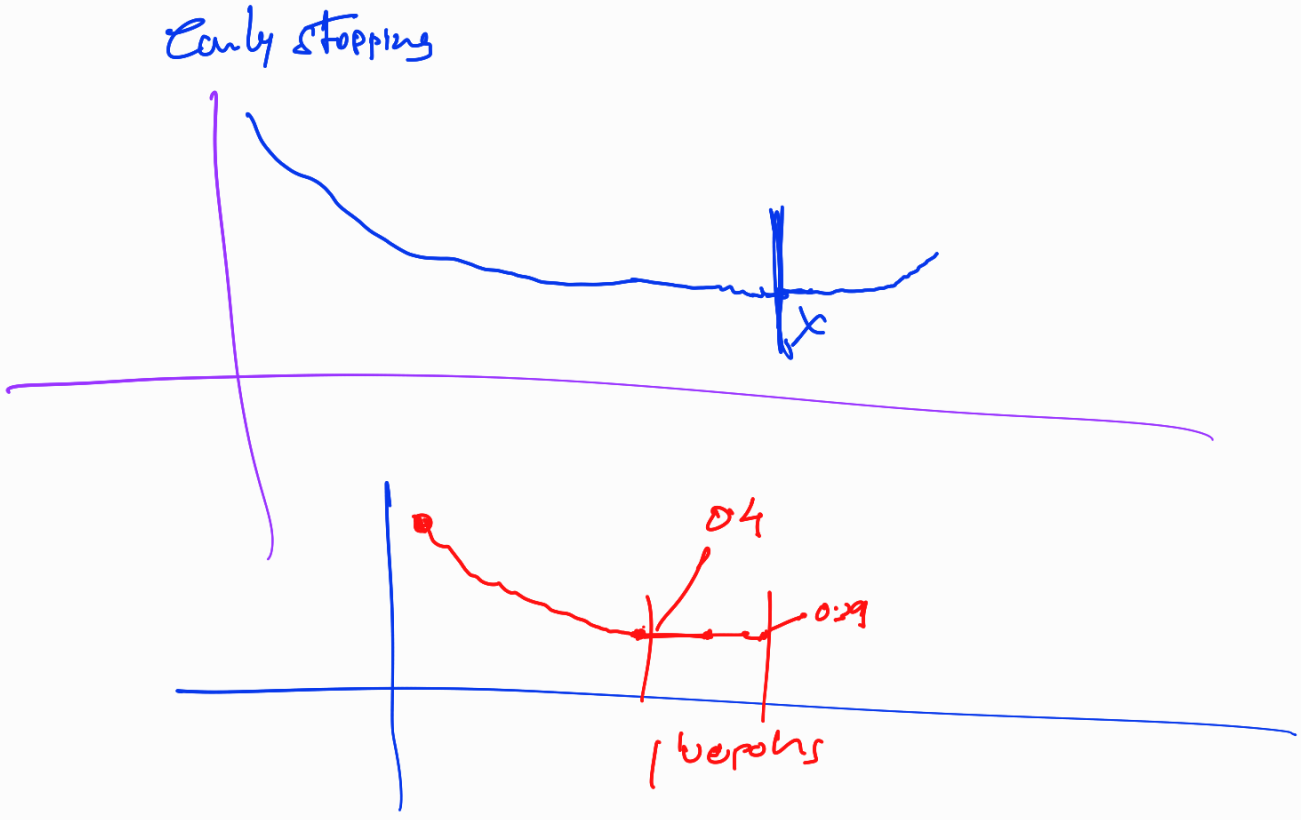


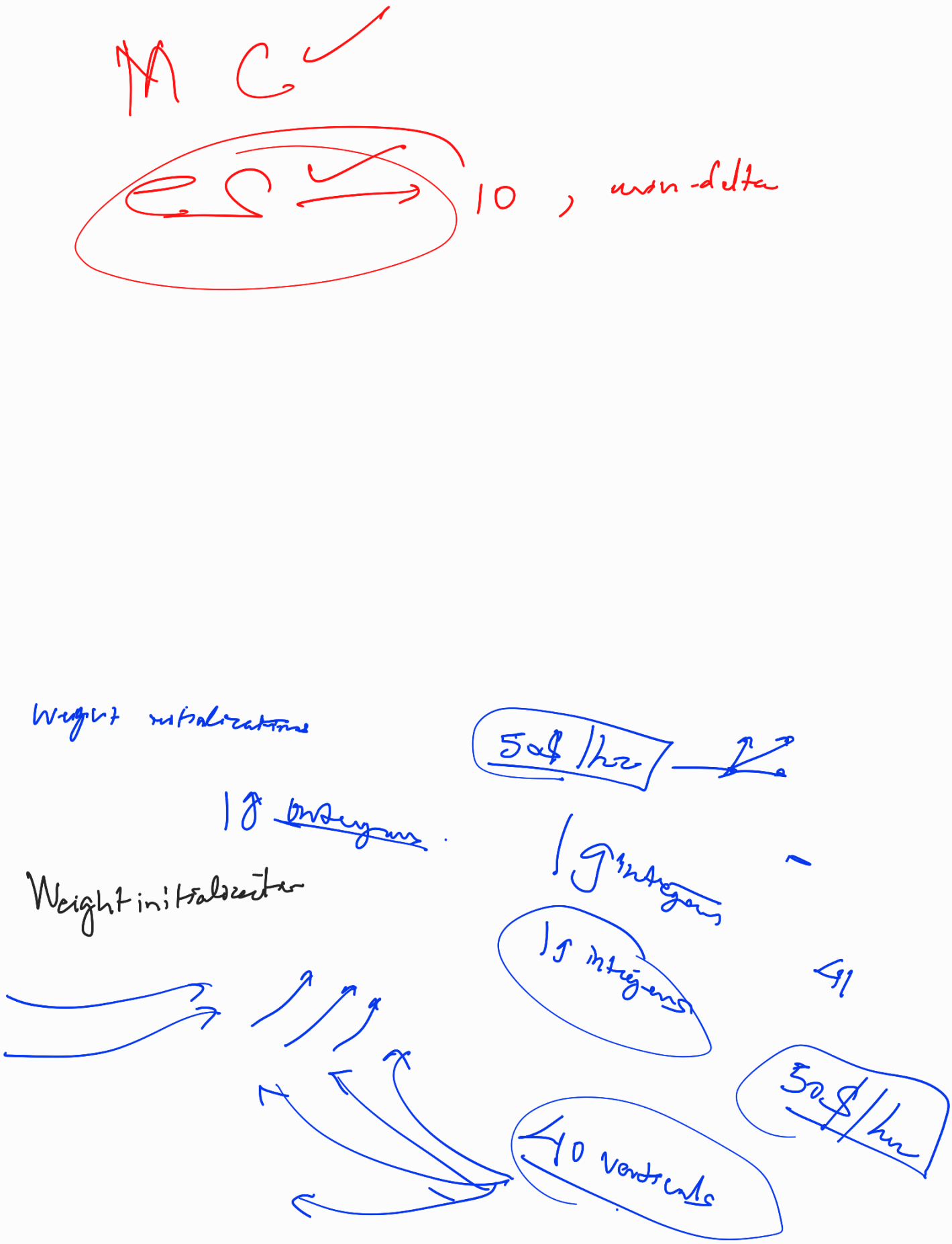
Epochs 60



~~model~~ Keras.load_model (file)

model.predict ( )




10 , max-delta
Weight initialization
50$/hr
1$ Integers
Weight initialization
1$ Integers
1$ Integers
50$/hr
40 Vendors