

**LEARNING ML/DL
FROM UNIVERSITY**

ONLINE COURSES

FROM YOUTUBE

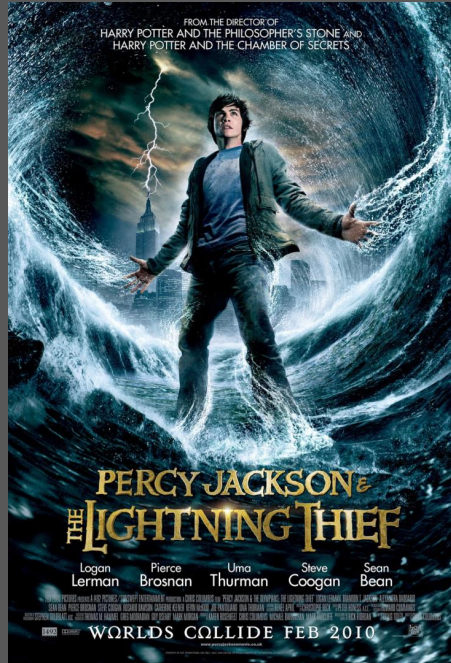
FROM ARTICLES

FROM MEMES

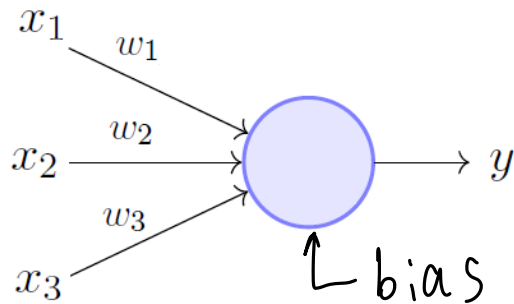


Discussion Cingin' in the Rain





Percy-epton



Perceptron Model (Minsky-Papert in 1969)

y
↓

$$\text{Output} = \text{Activation}(w_1 x_1 + w_2 x_2 + w_3 x_3 + \text{bias})$$

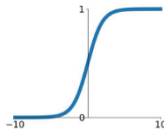


(BI)ac(k)-tivation Functions

Activation Functions

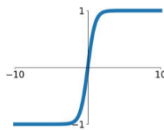
Sigmoid

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



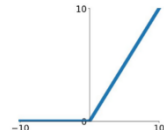
tanh

$$\tanh(x)$$



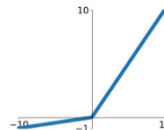
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

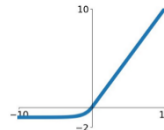


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Let's work on an X-ample



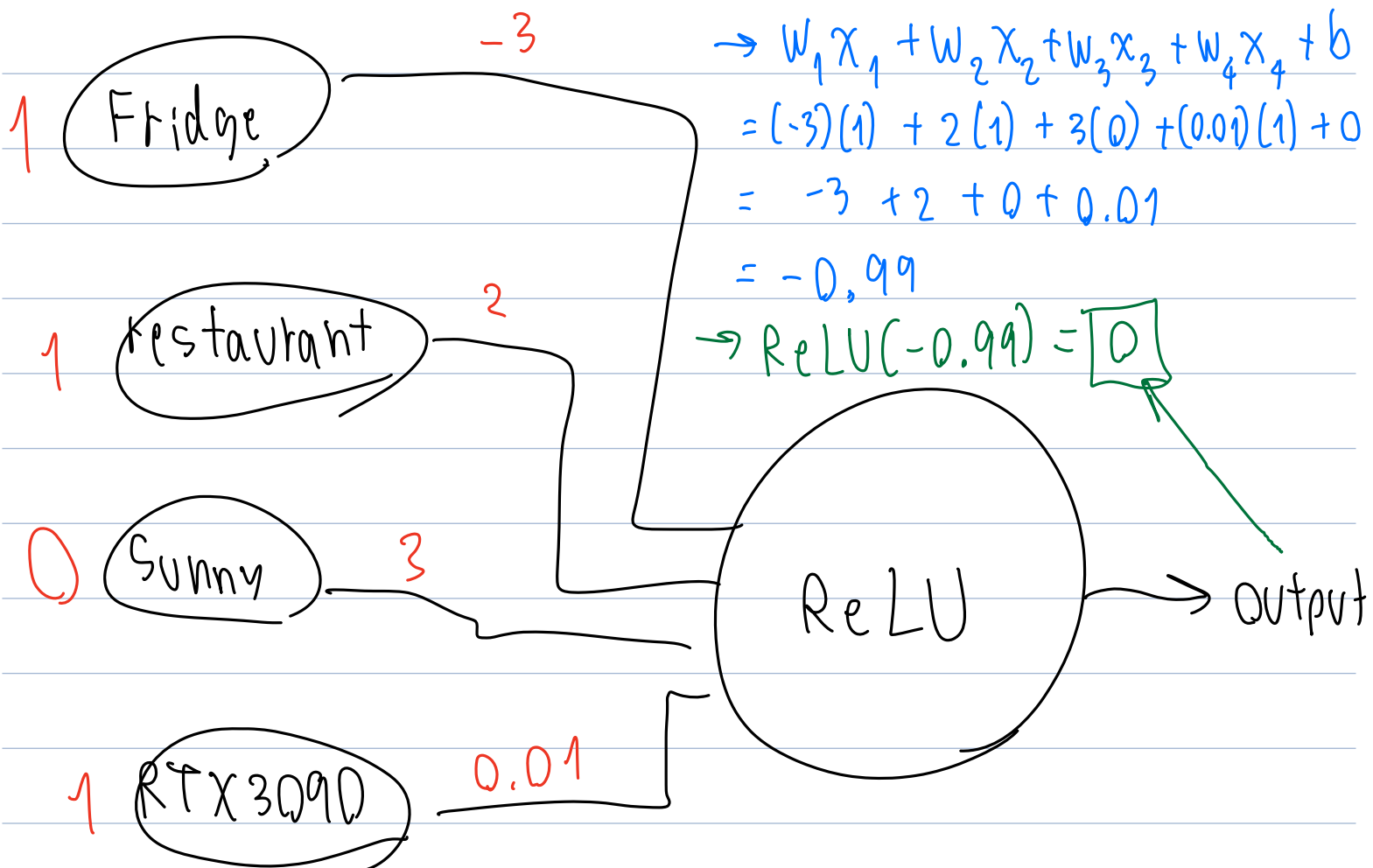
Ex Decide whether to eat outside? (positive number = Eat outside, 0 = No)

1) Anything left in the fridge (1 = Yes)

2) Any restaurant nearby? (1 = Yes)

3) Is it sunny now? (1 = clear sky)

4) Using RTX 3090? (1 = Yes)



Let bias = 0, activation func is ReLU

(A) New (Hope) Rai Network



Long Definition (optional):

Artificial neural networks, usually simply called neural networks (NNs), are based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal to other neurons. An artificial neuron that receives a signal then processes it and can signal neurons connected to it.

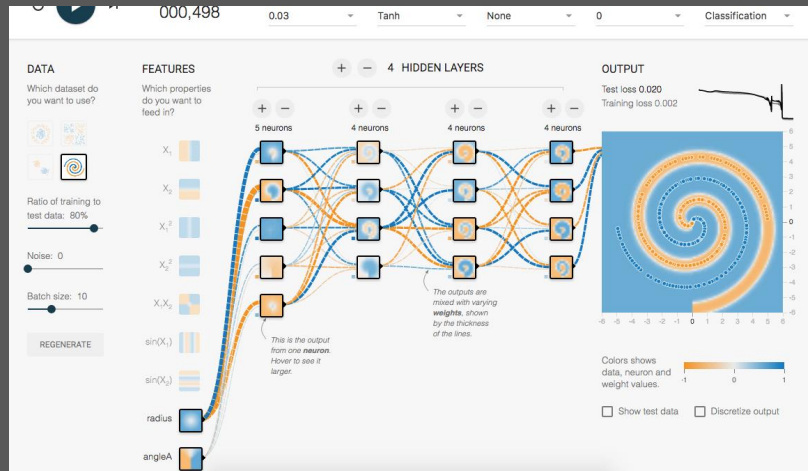
Short Definition:

Neural Network is a bunch of perceptron connected together

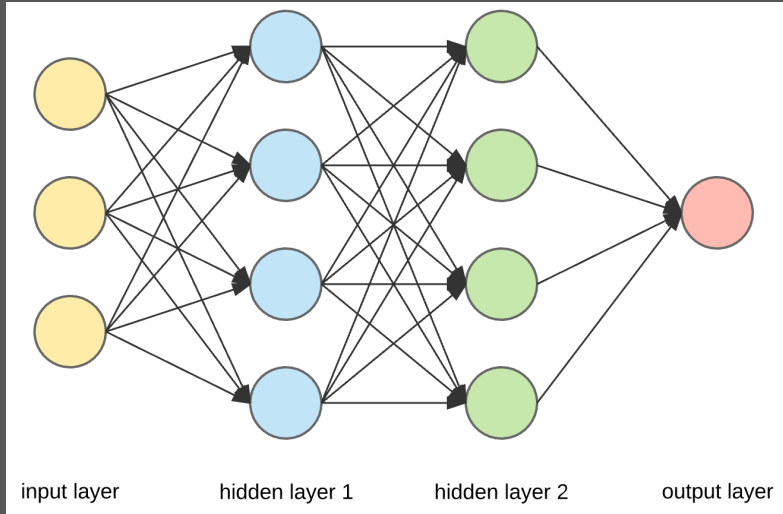
Motivation:

The perceptron is not complex enough to classify non-linear data. We then introduced neural network which consists of many perceptron to handle non-linear classification problem.

<https://playground.tensorflow.org/>



Definition & Terminology



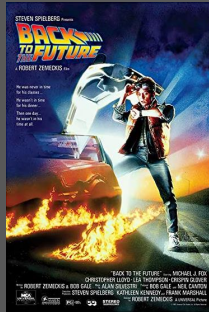
↑
input

↑
hidden

↑
output

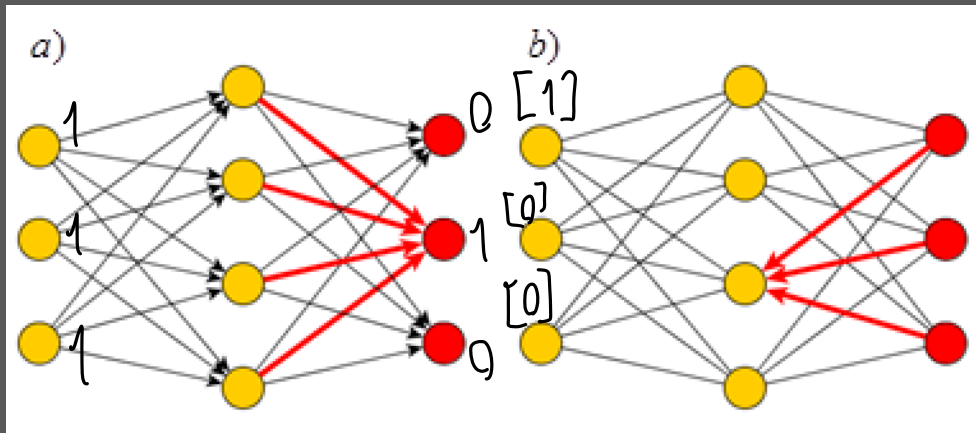
(Demon S)Layers





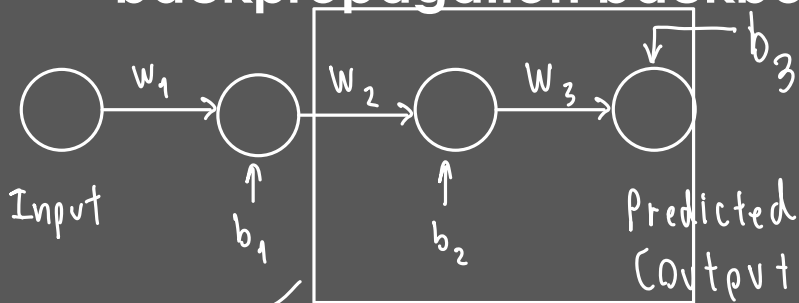
Back Propagation (backpropagation)

chain rule // given the cost value, we use
to update each weight & bias from output to
input layer



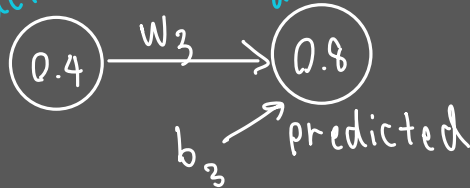


(Django Un)chain(ed) rule is
backpropagation backbone

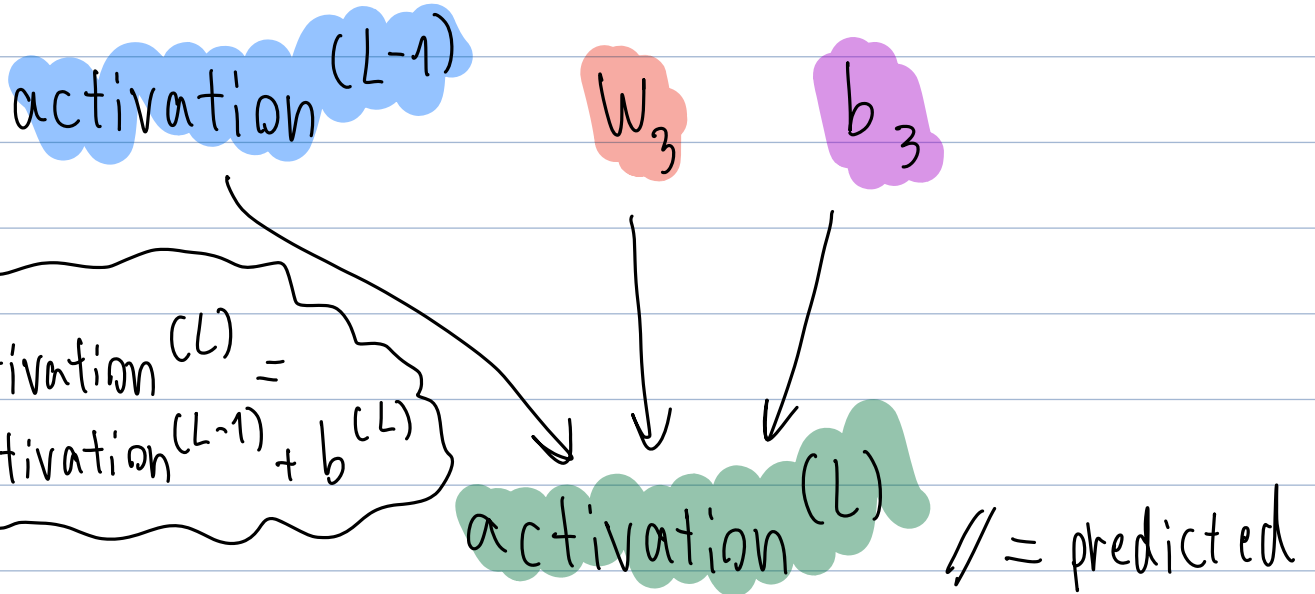
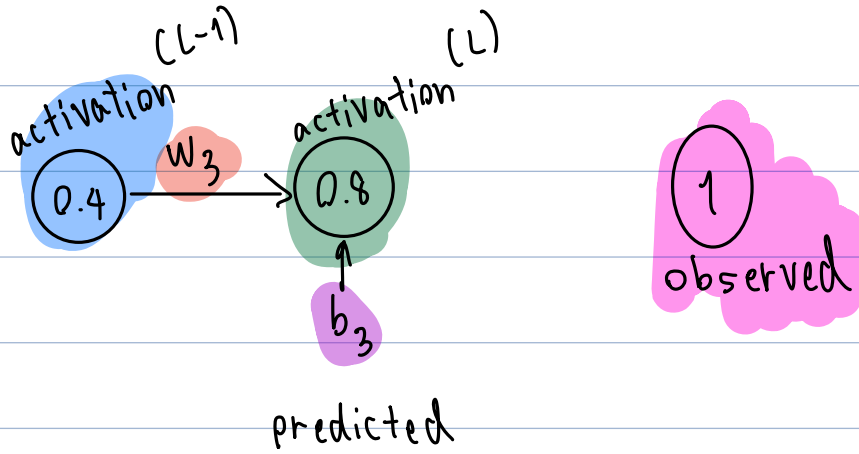


activation (L-1)

activation (L)



1
observed



Note: $\text{activation}^{(L)} = w^{(L)} \times \text{activation}^{(L-1)} + b^{(L)}$

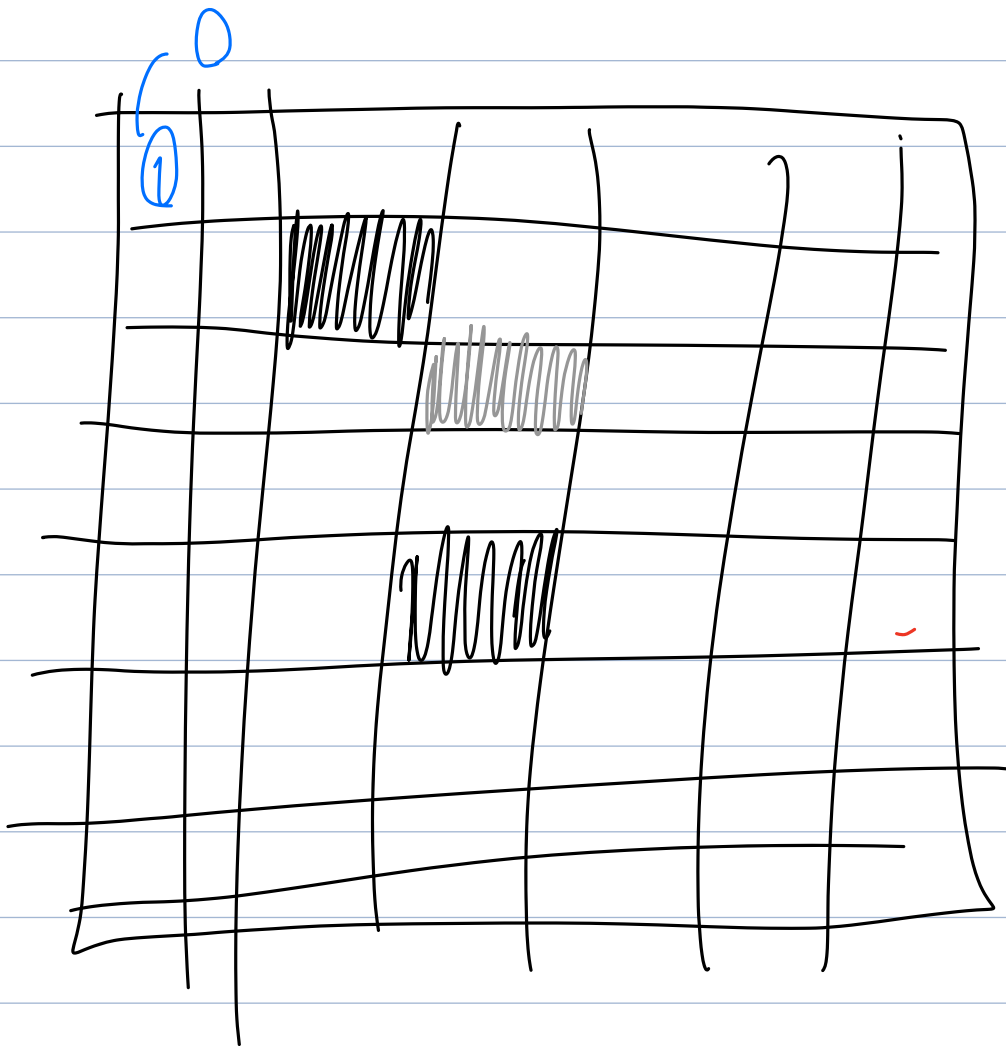
$$\begin{aligned} \text{Cost} &= (\text{predicted} - \text{observed})^2 \\ &= (\text{activation}^{(L)} - \text{observed})^2 \\ &= \# \end{aligned}$$

Gradient Descent: $w_{3\text{new}} = w_{3\text{old}} - \lambda \frac{\partial (\text{cost})}{\partial w_3}$

back @ That's The End (of Evangelion) for today

2:00 PM





BW \rightarrow $[0, 255]$
 B W

$(1, 1)$ 0 \leftarrow value 0
 $(1, 2)$ 0 \leftarrow 100
 $(1, 3)$ 0 \leftarrow 222
 \vdots
 \vdots
 \vdots

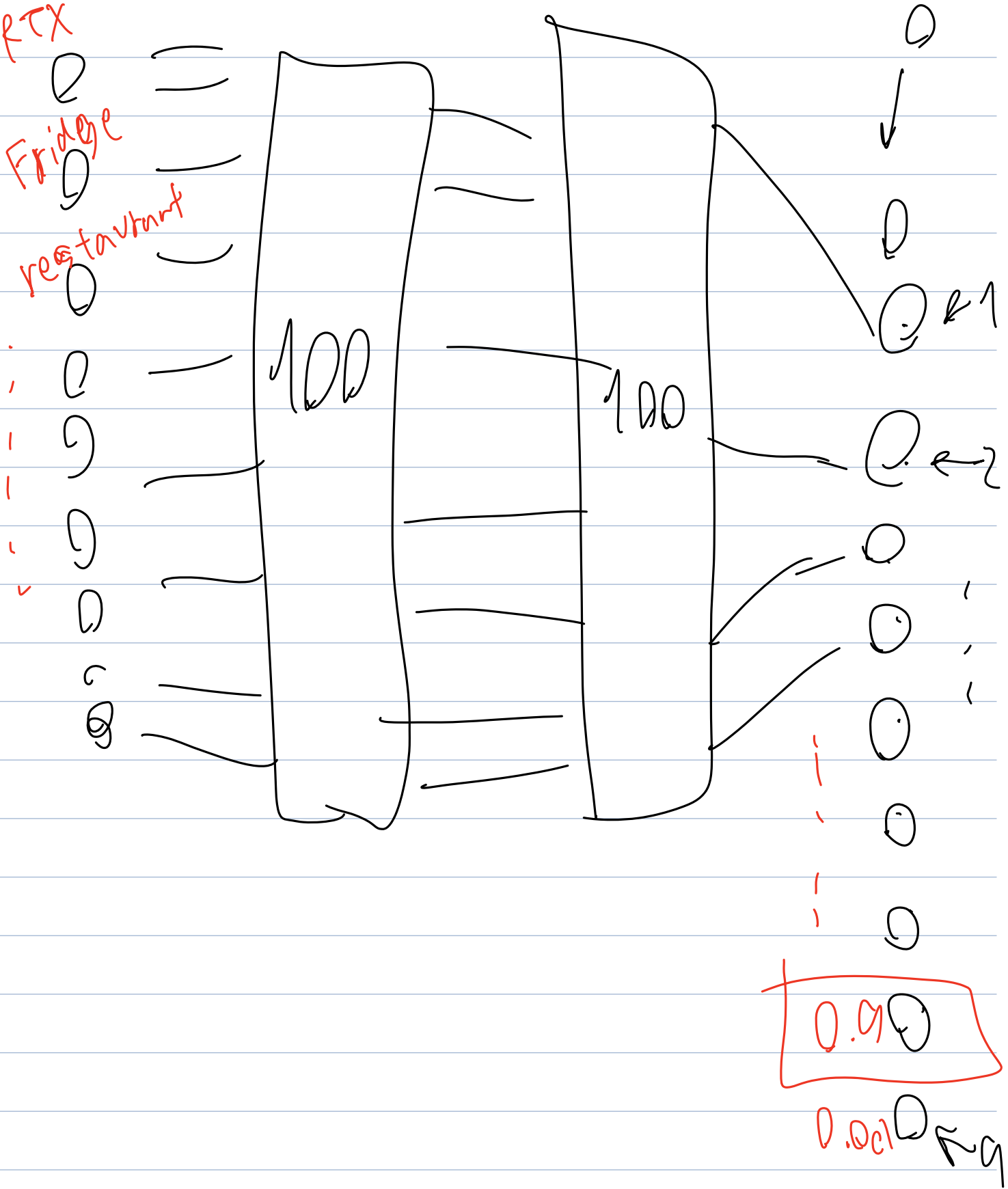
MLP \rightarrow Multi-layer
 Perceptron

(NN)

RTX

Fridge

restaurant



"Dropout"

