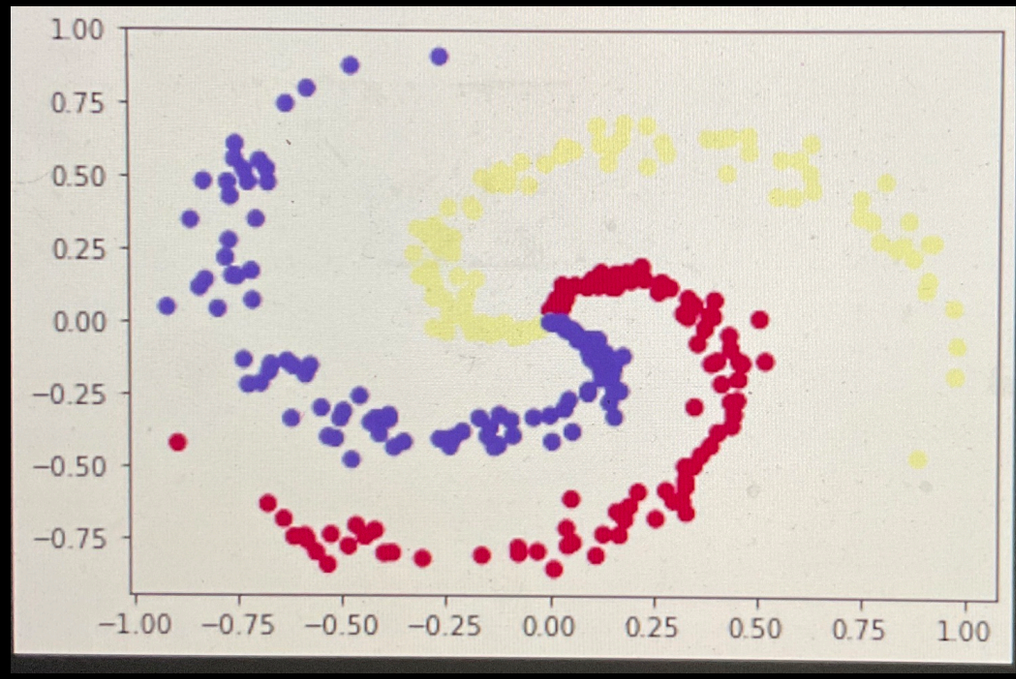
Introduction to Neural Networks

Problem Statement



Challenges with this data-set ?

→ Non Linear

→ Multi Class

Let's evaluate all ML algos' that we know.

→ LogReg → DT : RF, GBDT . . .

→ KNN → SVM → NB . .

1) Logistic Regression

Multiclass . — Default ✗
ovc -vs- rest ✓

Non- linear — Default ✗
Polynomial reg ✓

Other issues — Manual Feature eng.

2) KNN

Multiclass - ✓

Non-linear - ✓, sensitive to distance metric

Other issues - Curse of dimensionality

"Non parametric" model

3) Tree based models

Multiclass - ✓

Non-linear -  → Zigzag

Other issues - Image / Text / Sparse
↓
High Dimensional
→ Trees fail here.

↳ SVM

Multiclass - ✗
One-vs-rest ✓

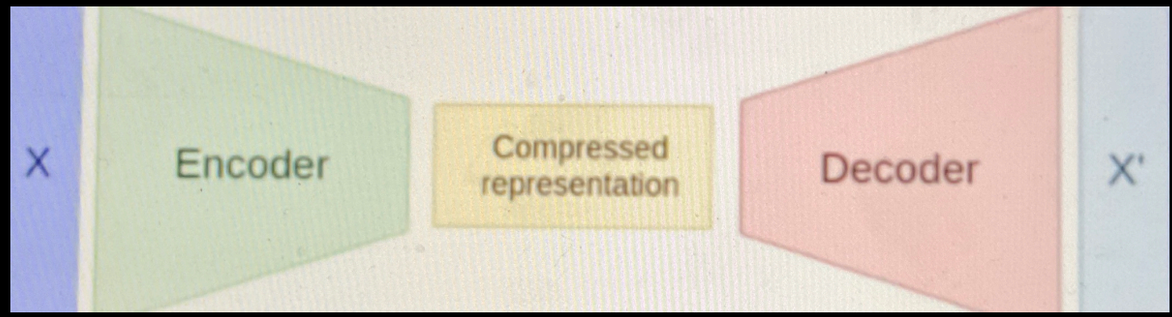
Non-linear - ✗
Kernel ✓

Other issues - Bad performance on large data sets (kernels are slow)

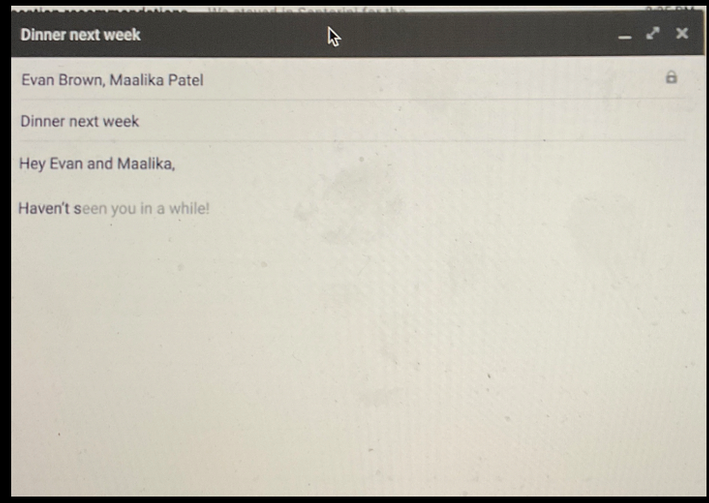
Neural Networks

Applications

Image Segmentation

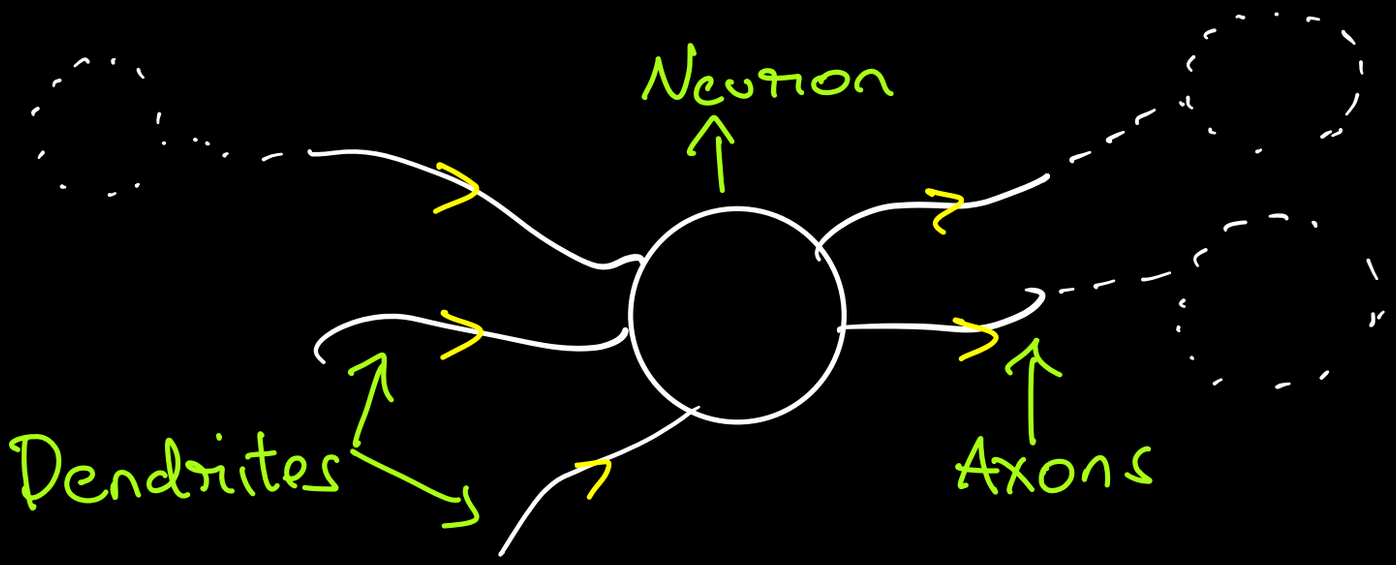


Compression



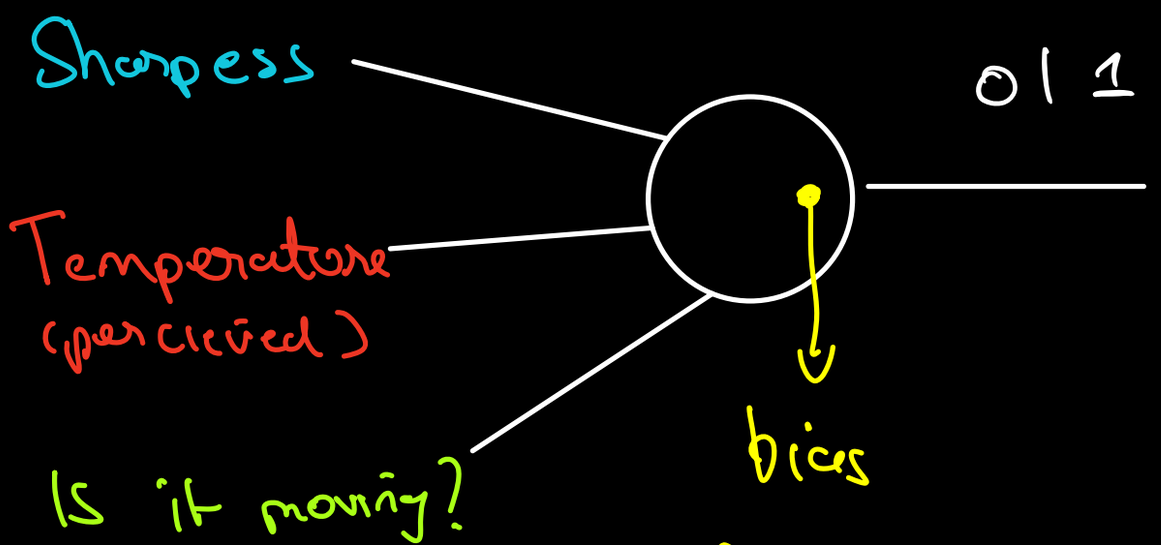
Auto Complete

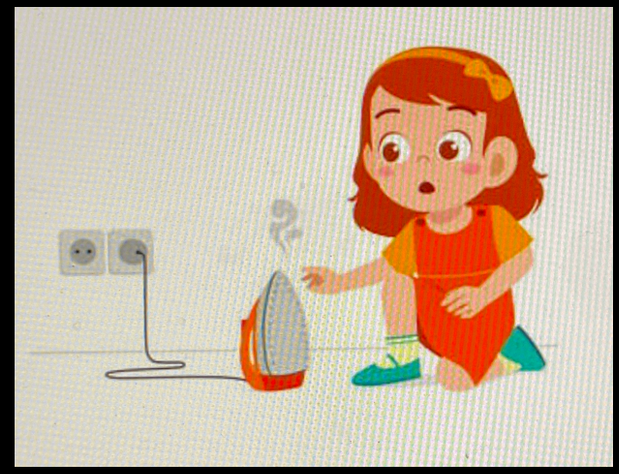
Biological Neural Network



Task: Touch or not touch ?

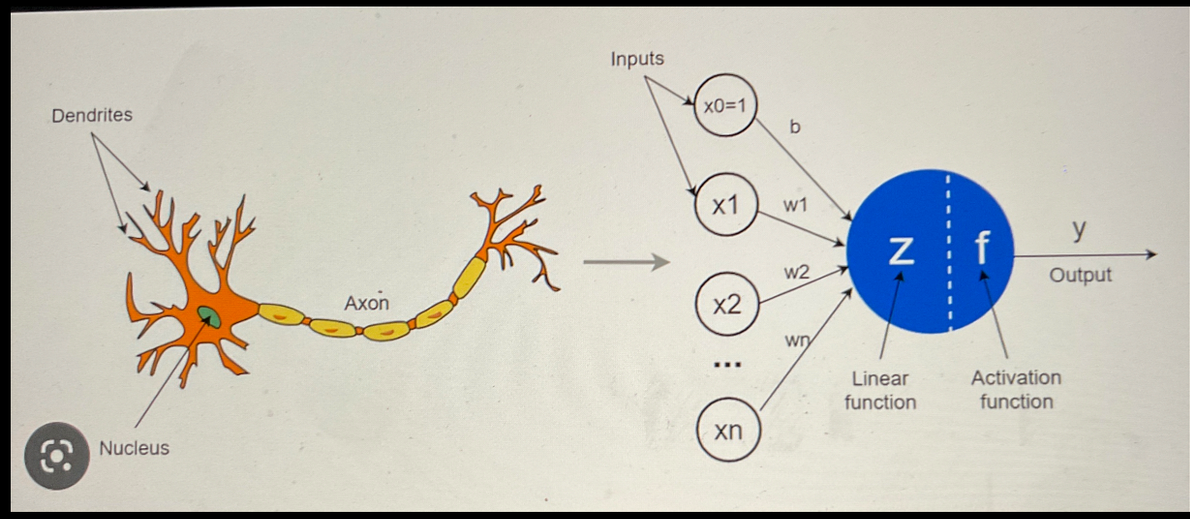
Biology: Neuron → Decide to touch ...



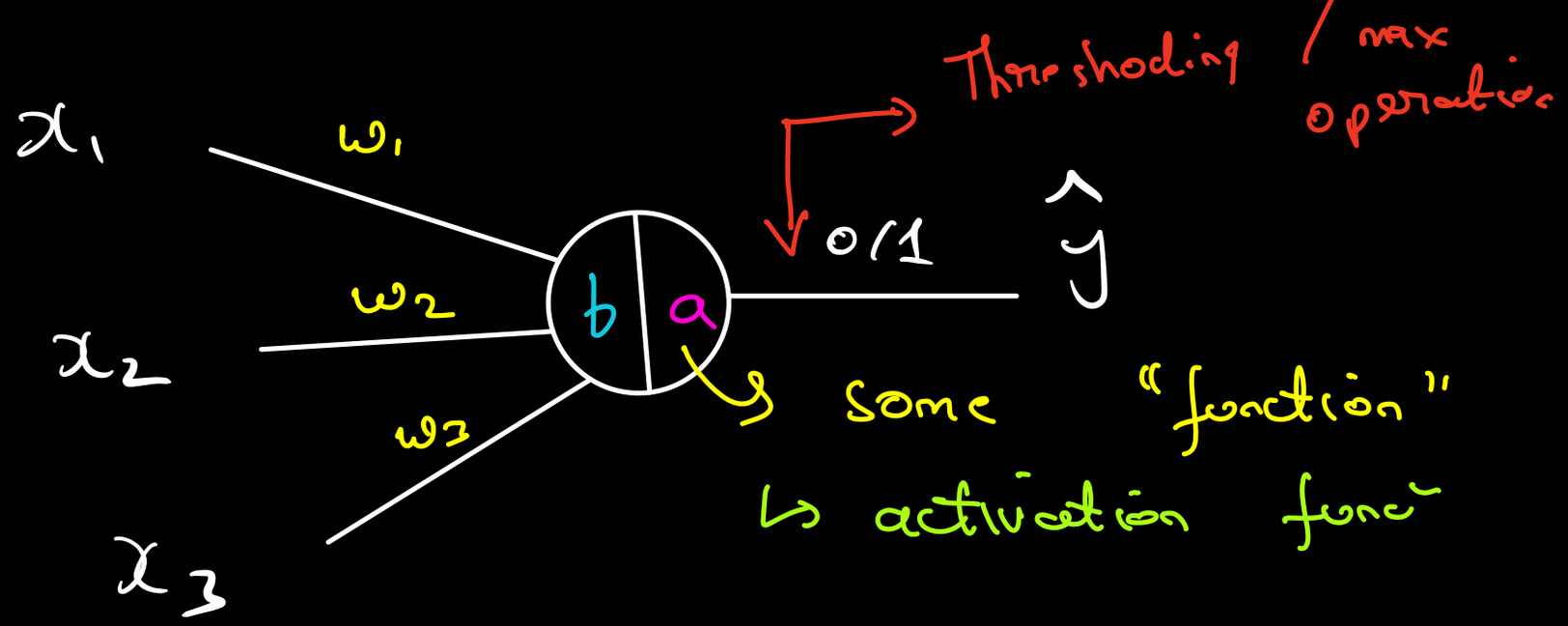


(prior experience)

The weights of input information can be different.



An Artificial Neuron

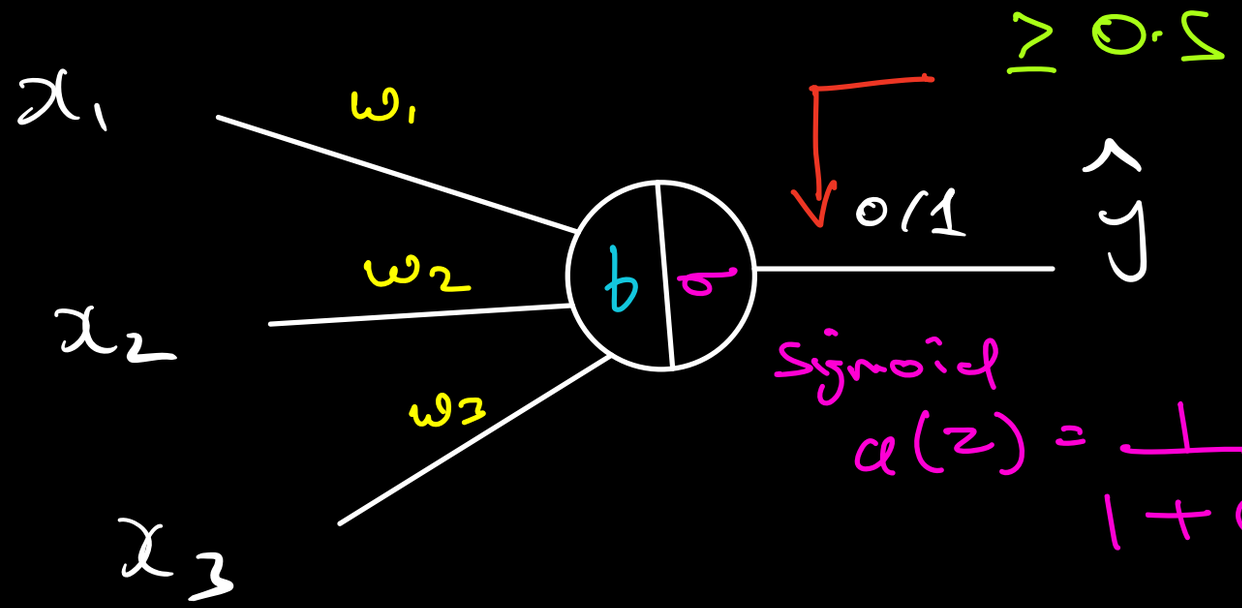


$$\hat{y} = a(w_1 x_1 + w_2 x_2 + w_3 x_3 + b)$$

A.K.A → **Perceptron**

Logistic Regression as a Neuron

a = Sigmoid



$$\hat{y} = \text{Sigmoid}(w^T x + b)$$

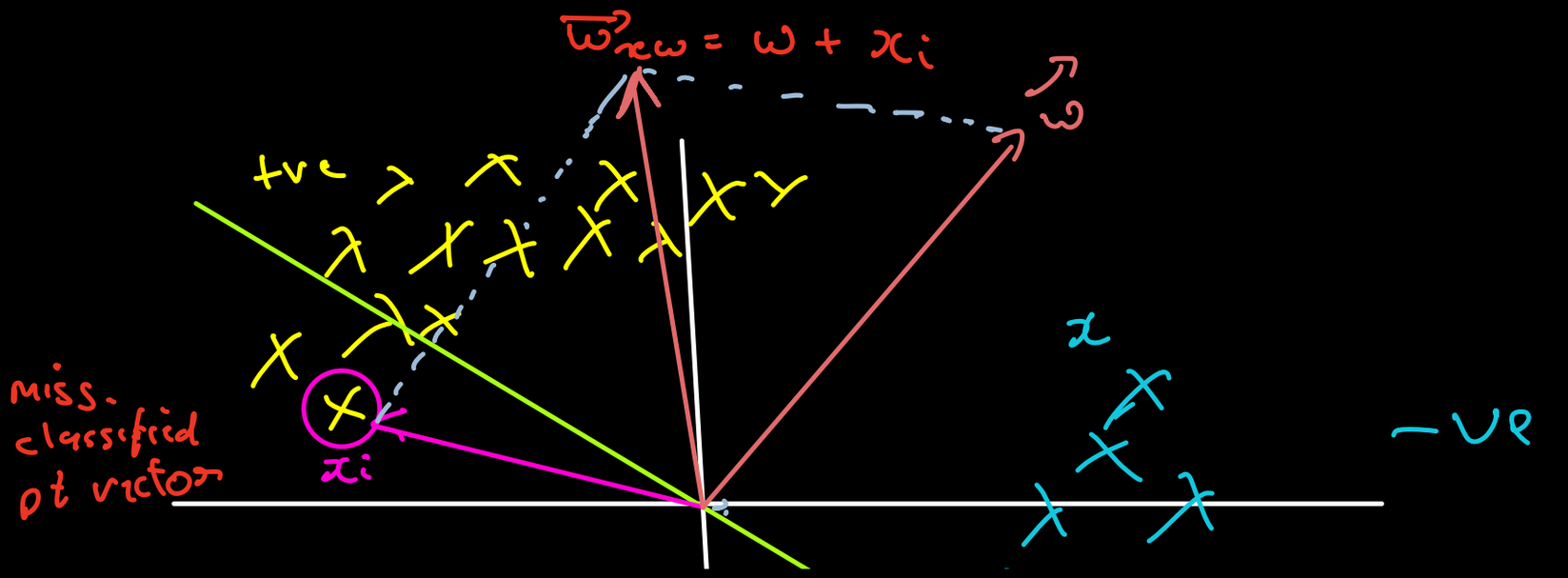
→ A.K.A **Logistic Regression Unit (LRU)**

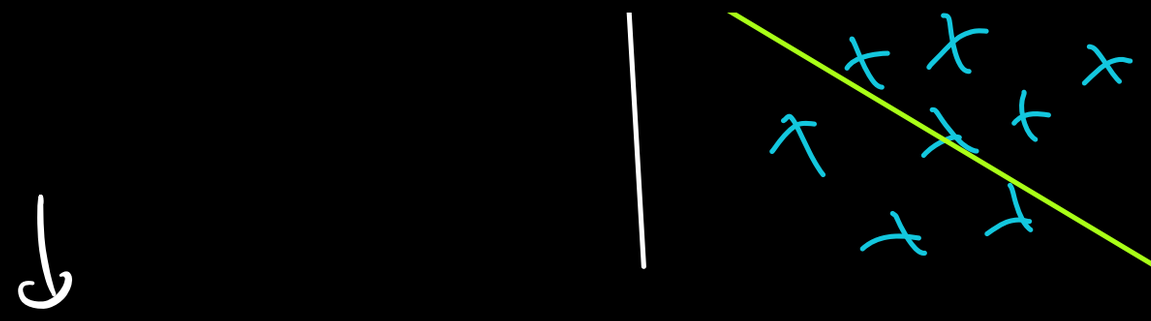
Similarly for

Linear Regression : $a(z) = z$

Perceptron

Remember the perceptron algo? (1969)

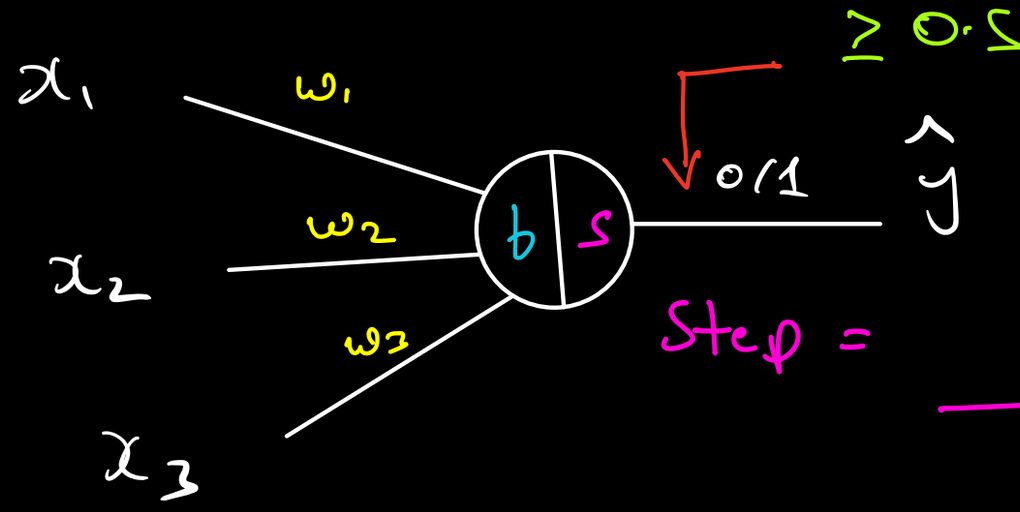




$w^T x = 0$

The way that this model takes decisions was →

$$\text{class} = w^T x + w_0 \begin{cases} > 0 \rightarrow \text{+ve HS} \\ < 0 \rightarrow \text{-ve HS} \end{cases}$$

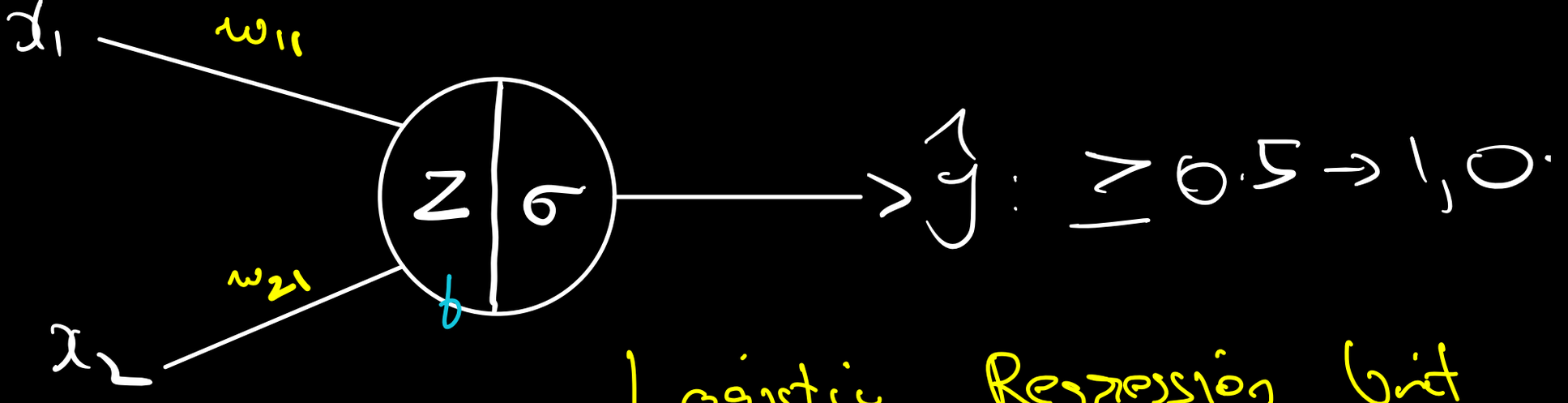


$$\text{Step} = \sqcap \quad a(z) = \begin{cases} 1 ; & z \geq 0 \\ -1 ; & z < 0 \end{cases}$$

From 'Neuron' to "Network"

Agenda:

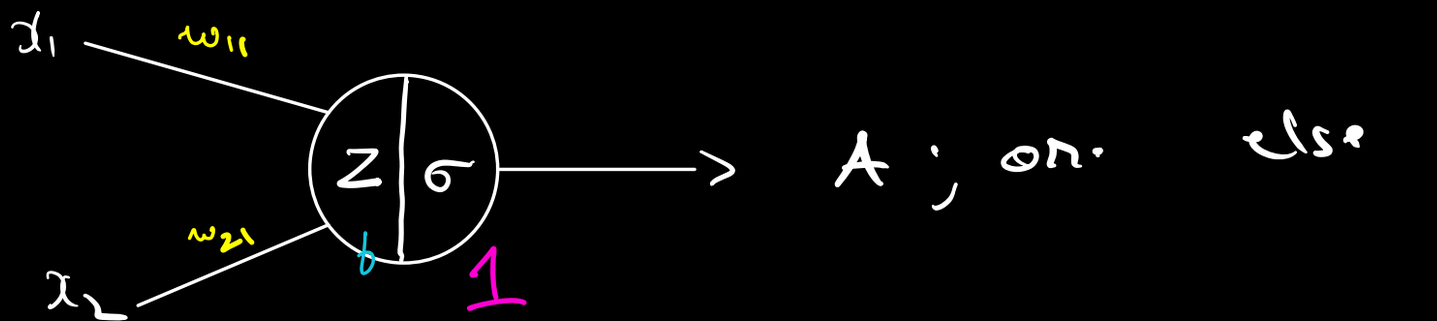
Make Logistic Regression support multi-class

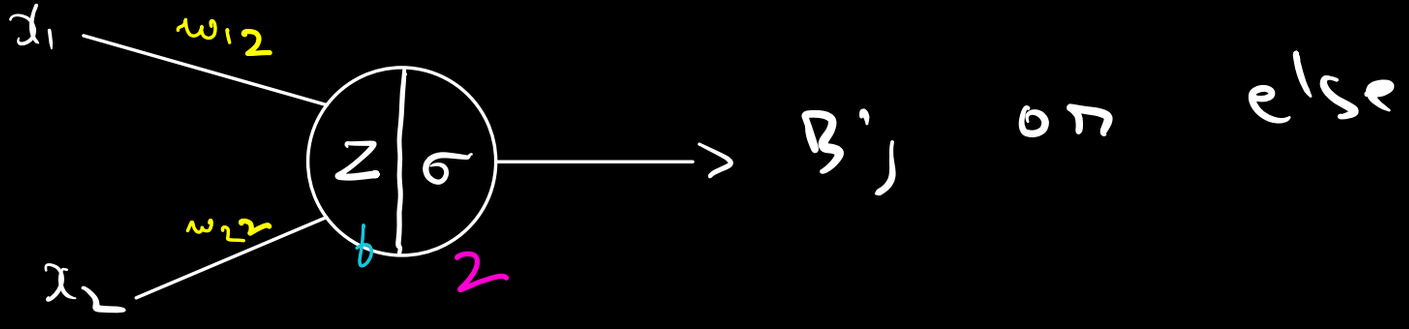


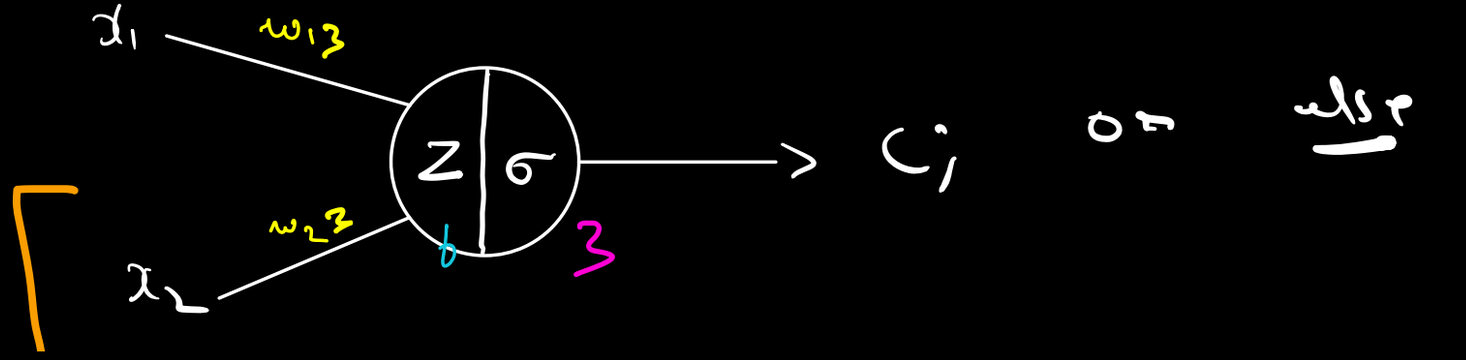
Logistic Regression Unit

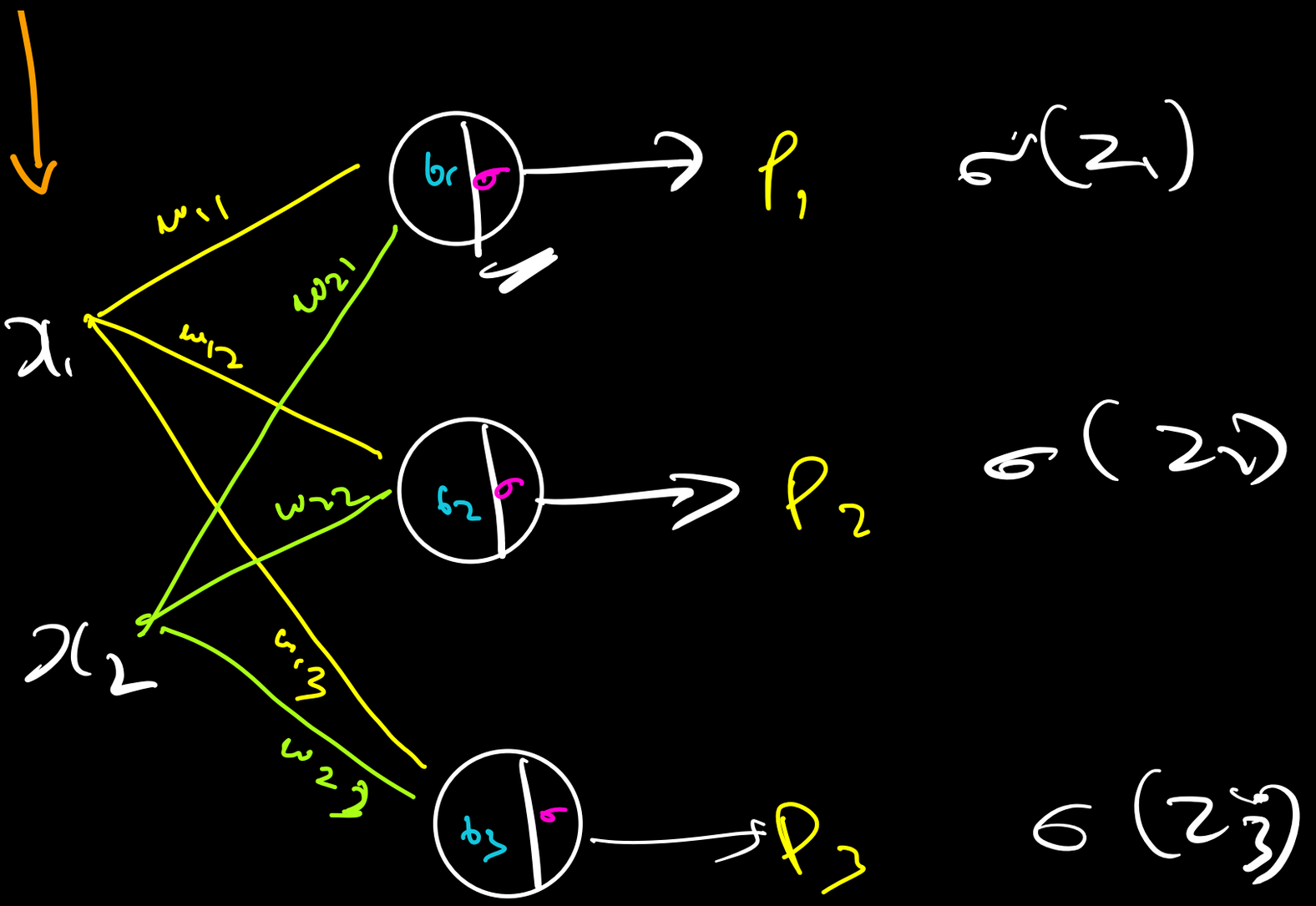
[LRU]

Multi Class One vs Rest



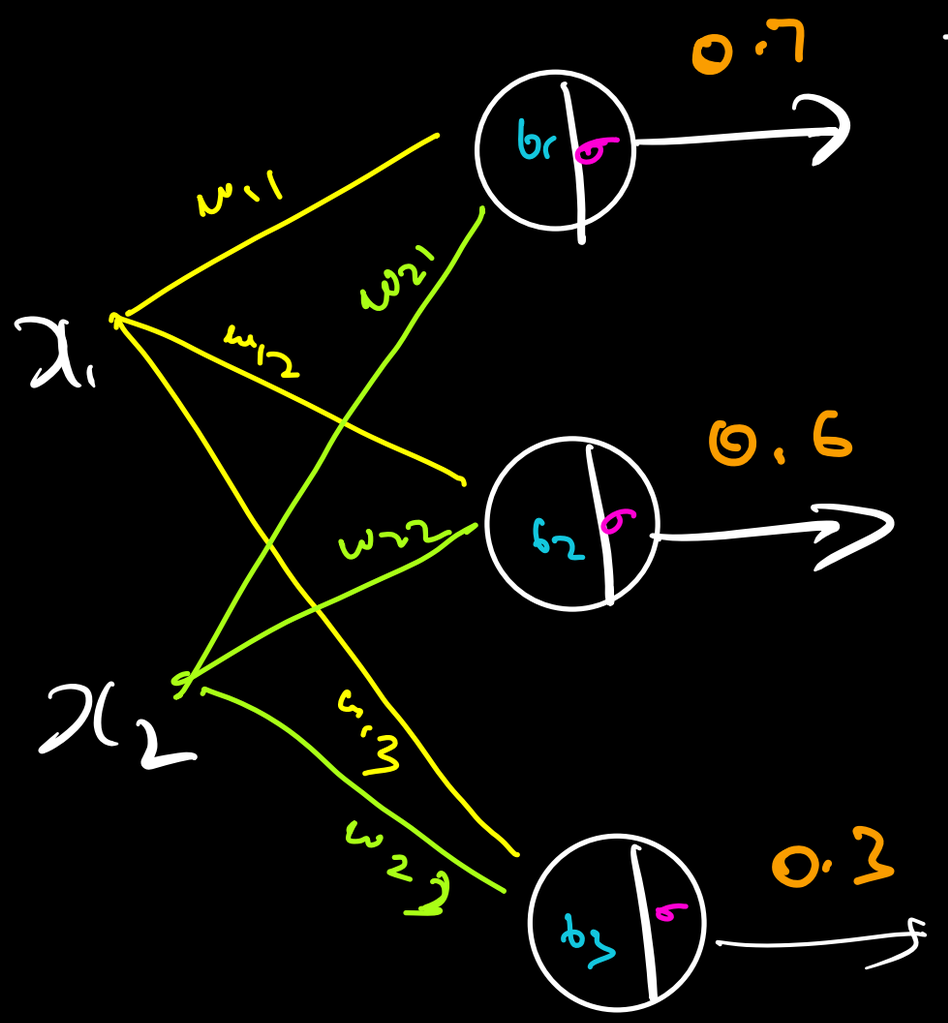






Is there any difference b/w this page and previous page? → NO

Note that you need to one-hot-encode y for this to work



y	y
1	1
1	0
0	0

Notice this is sum of prob ≥ 1

Solution ?

→ Normalize !

$$\text{New } P_i = \frac{P_i}{P_1 + P_2 + P_3} = \frac{P_i}{\sum P_j}$$

Hence we can define a new activation func^n using → $P_i = \sigma(z_i)$

$$a_{new}(z_i) = \frac{\sigma(z_i)}{\sum_j \sigma(z_j)}$$

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

$$a_{new}(z_i) = \frac{\frac{e^{z_i}}{1+e^{z_i}}}{\sum_j \frac{e^{z_j}}{1+e^{z_j}}}$$

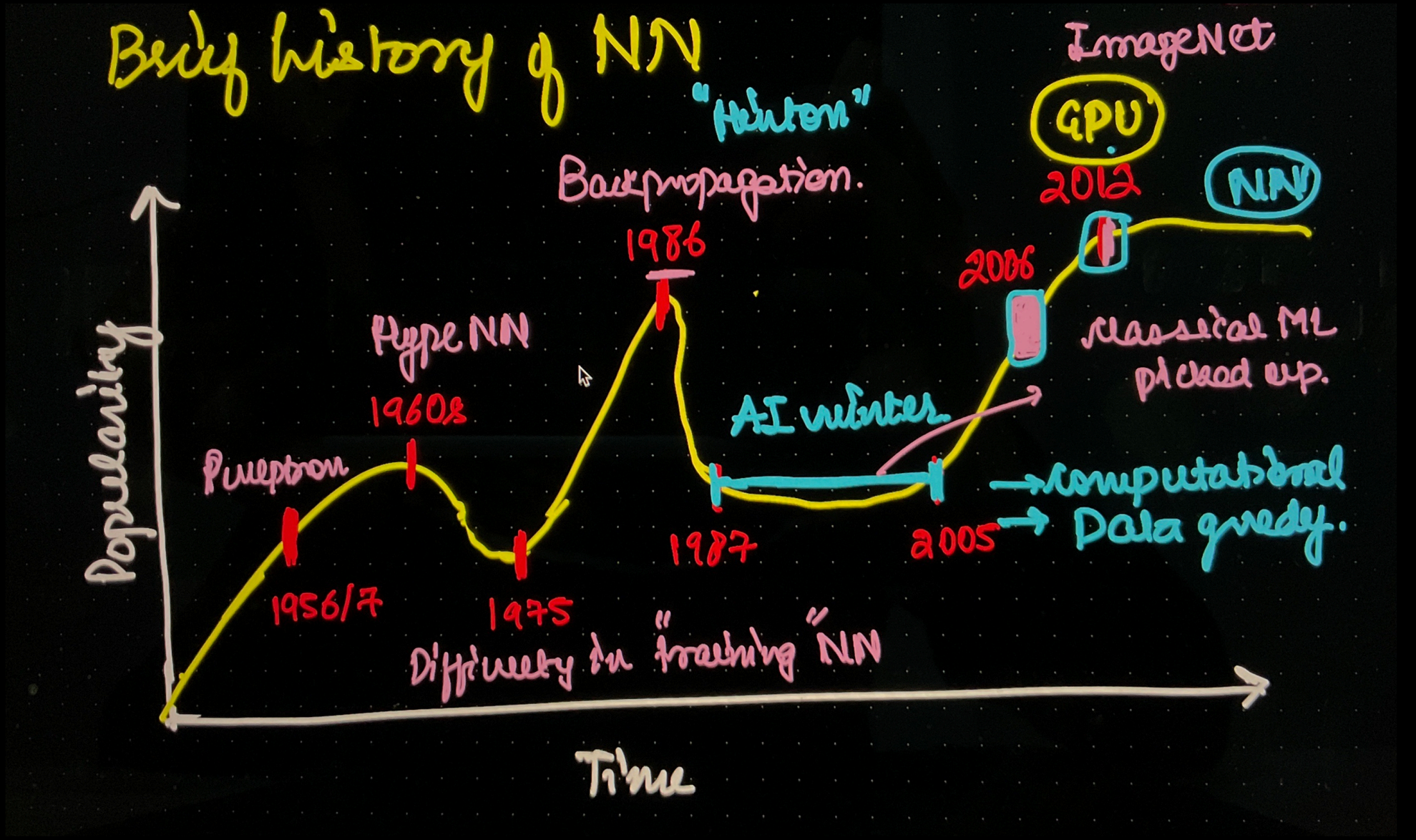
→ Remove

→ Remove

This is a valid idea. However we can make it simpler and define a new activation function

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

$$\therefore p_1 = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}}$$


Brief history of NN
"Hinton"
Backpropagation.
1986
ImageNet
GPU
2012
NN
2006
Hype NN
1960s
Perceptron
1956/7
1975
AI winter
1987
2005
classical ML
picked up.
→ computational
→ Data greedy.
Difficulty in "training" NN
Popularity
Time

What has changed recently?

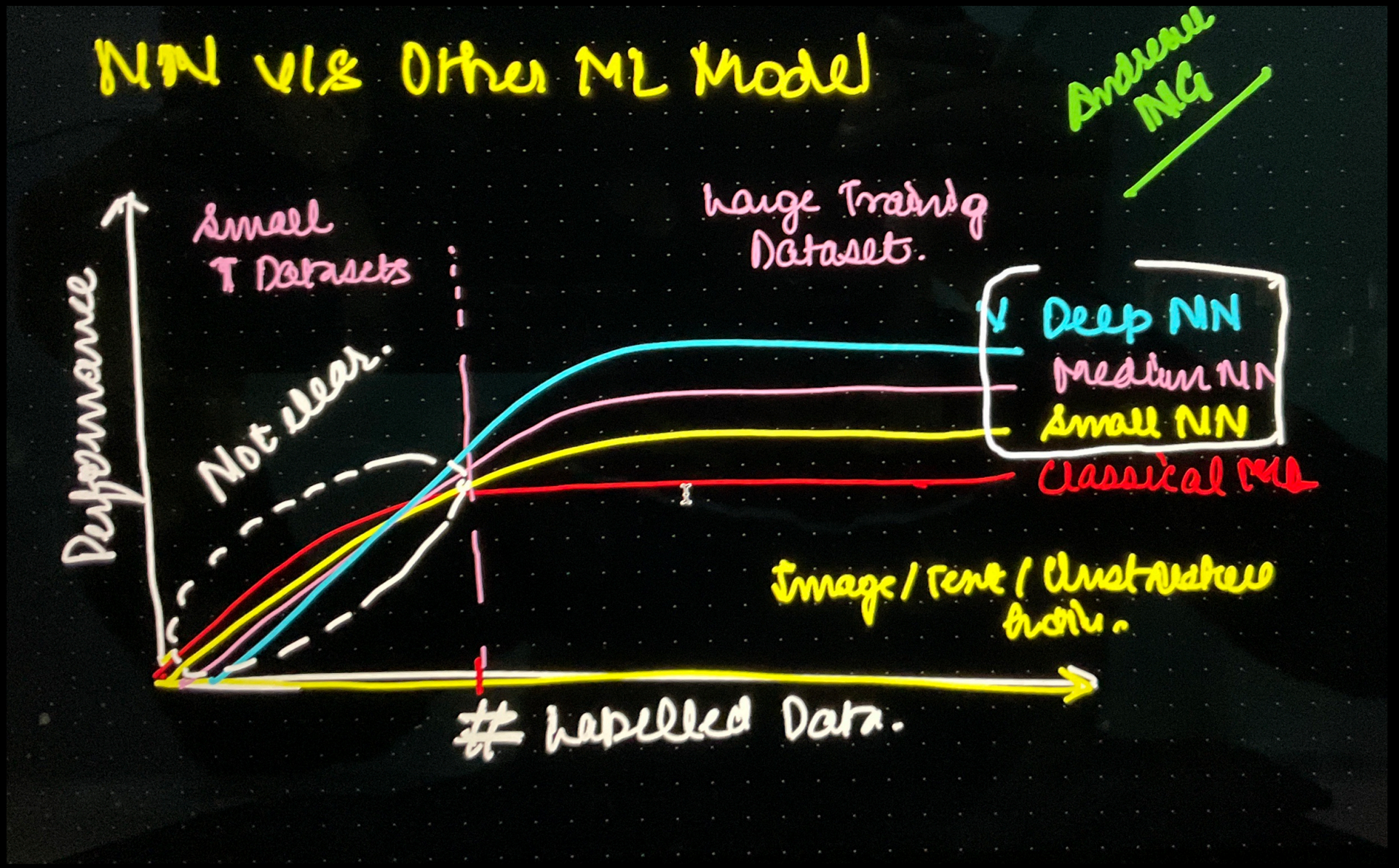
Why AI winters?

→ Lack of computational power.
→ Lack of data

NN V/S DL

1. Comp. Power. → GPU/TPU/ Neural Engine.
2. Data ← Sensor, IOT
3. Research on newer training, new algos, newer loss fn.

"AI winters" - 'Re-brand' - Deep Learning.
NN Training deep NN.


NN vis Other ML Model
Andrew NG
Small Datasets
Not clear.
Large Training Dataset.
Deep NN
Medium NN
Small NN
Classical ML
Performance
Image / Text / Unstructured data.
Labelled Data.