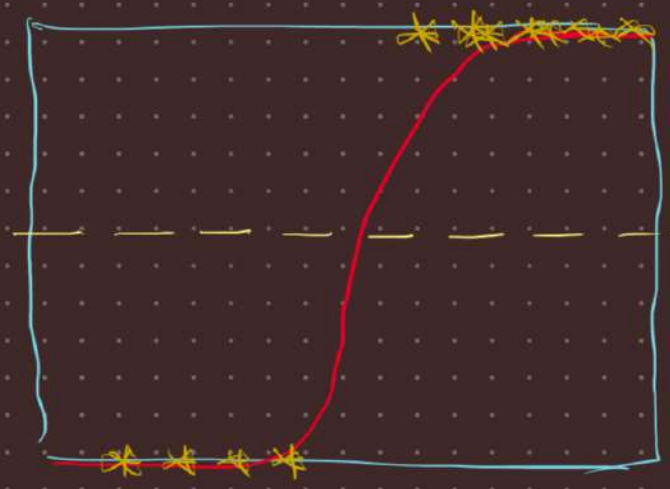
ARTIFICIAL NEURAL NETWORK

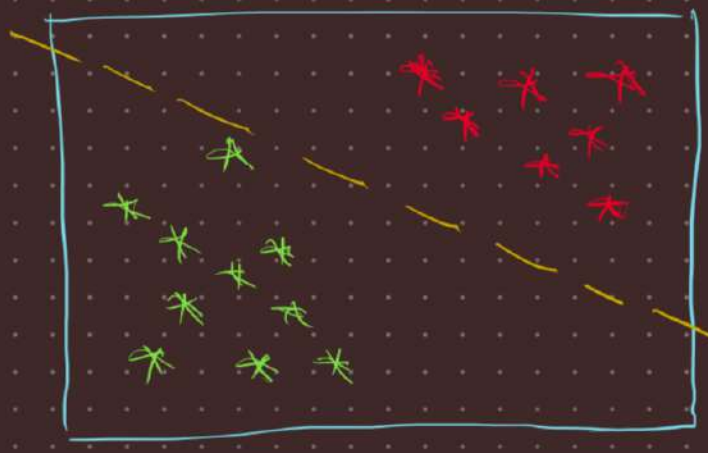
This algorithm has changed the whole phase of ML

Sub theories

(1) Every Logistic Regression line gives us a decision boundary. It looks like a straight line between Class 0 and Class 1.



Logistic Reg line gives us a decision boundary



(2) New representation for Logistic Regression

We are trying to find weights by minimizing error. While building any model,

$\beta_0 = w_0$

LOGISTIC REGRESSION LINE

$$y = \frac{e^{(\beta_0 + \beta_1 x_1 + \beta_2 x_2)}}{1 + e^{(\beta_0 + \beta_1 x_1 + \beta_2 x_2)}} \rightarrow (1)$$

ANN REPRESENTATION

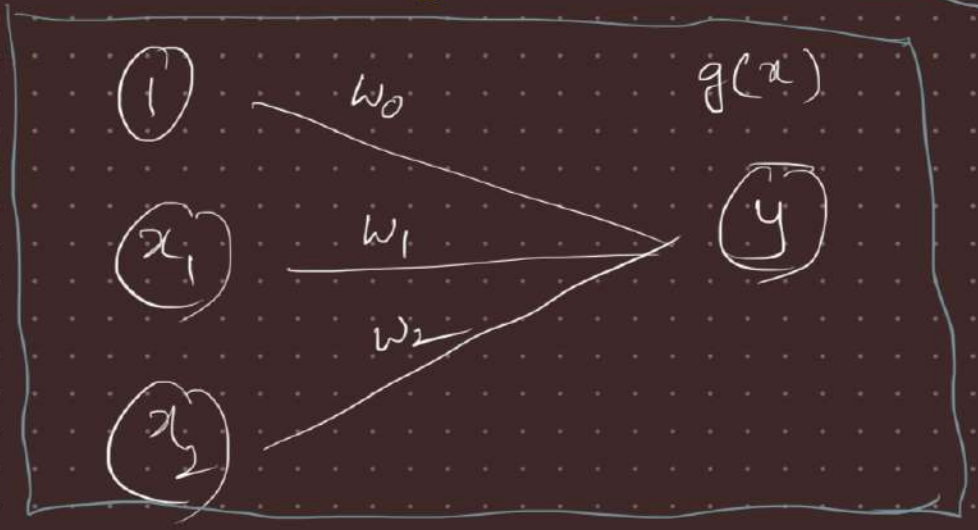
$$y = \frac{e^{(w_0 + w_1 x_1 + w_2 x_2)}}{1 + e^{(w_0 + w_1 x_1 + w_2 x_2)}} \rightarrow (2)$$

ANN REPRESENTATION

$$y = g(w_0 + w_1 x_1 + w_2 x_2) \rightarrow (3)$$

where $g(x) = \frac{e^x}{1 + e^x}$

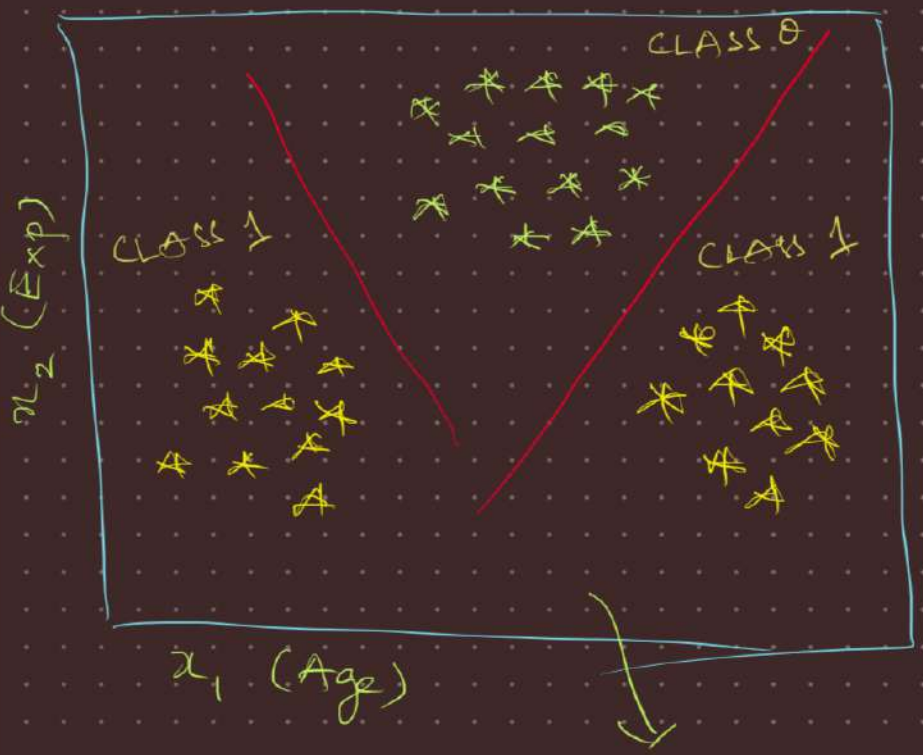
ANN REPRESENTATION



Note: Logistic Reg and ANN are one and the same.

(3) Logistic Regression Line fails in case of multiple/non linear decision boundaries.

accuracy of model is very less

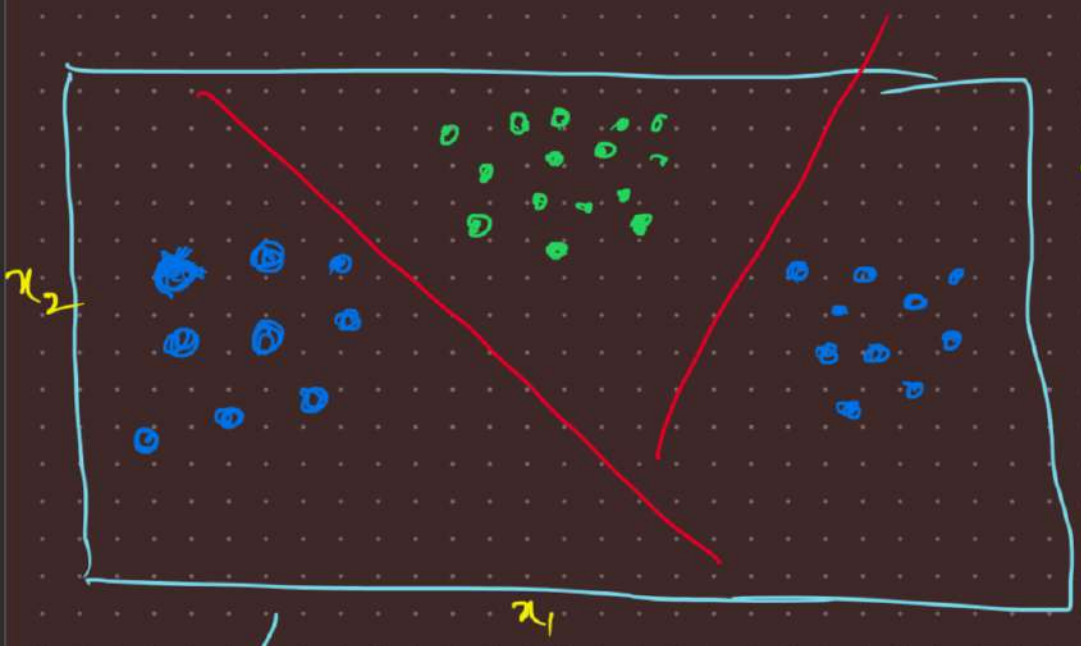


Age	Exp	y-pred
		0
		1
		0
		0
		1
		1

Multiple Decision Boundaries

This data is little complex

(4) We used intermediate outputs to solve the problem of non-linear decision boundaries

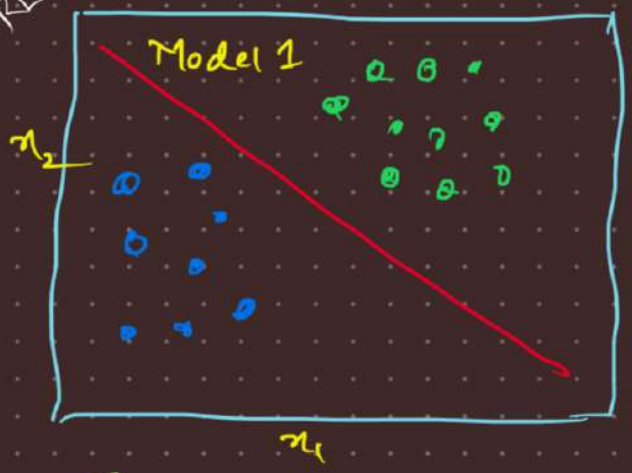


x_1	x_2	h_1	h_2	y

use h_1 & h_2 to predict y

REGION 1

Logistic Reg Model 1



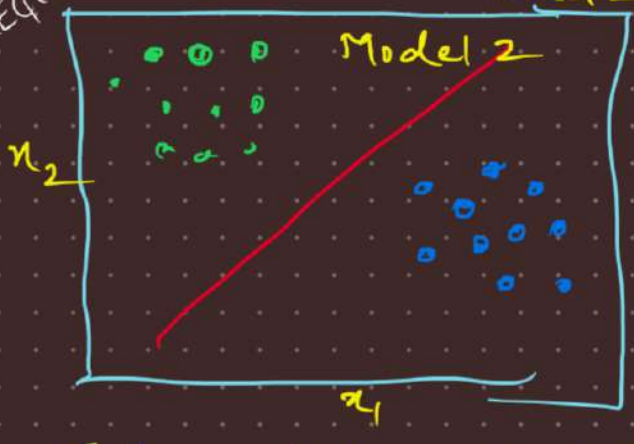
Intermediate output 1 (h_1)

↓

predicted values from Model 1

REGION 2

Logistic Reg Model 2

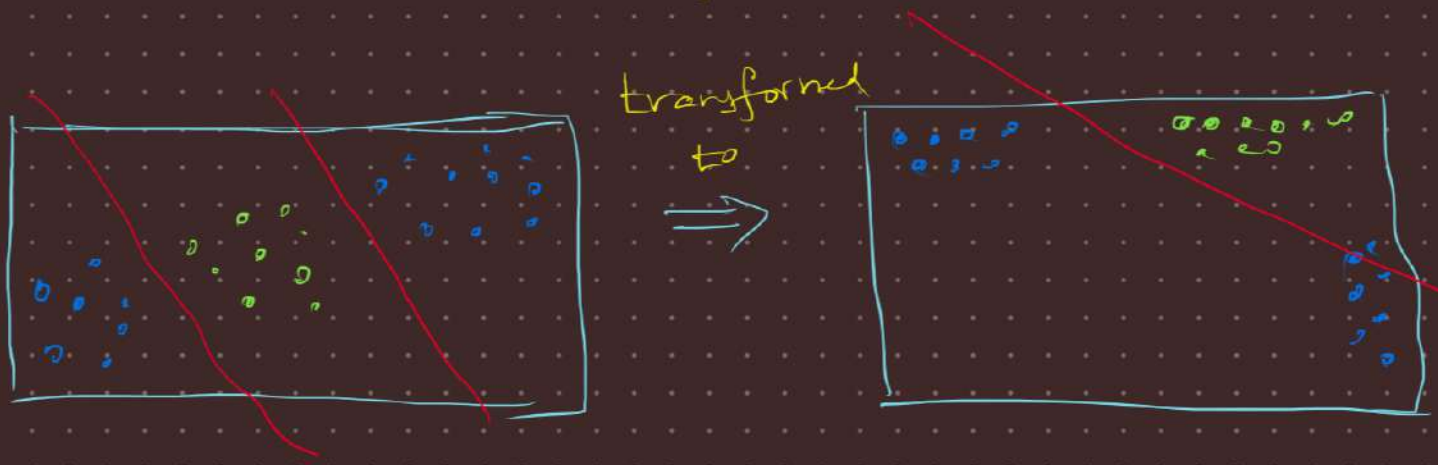


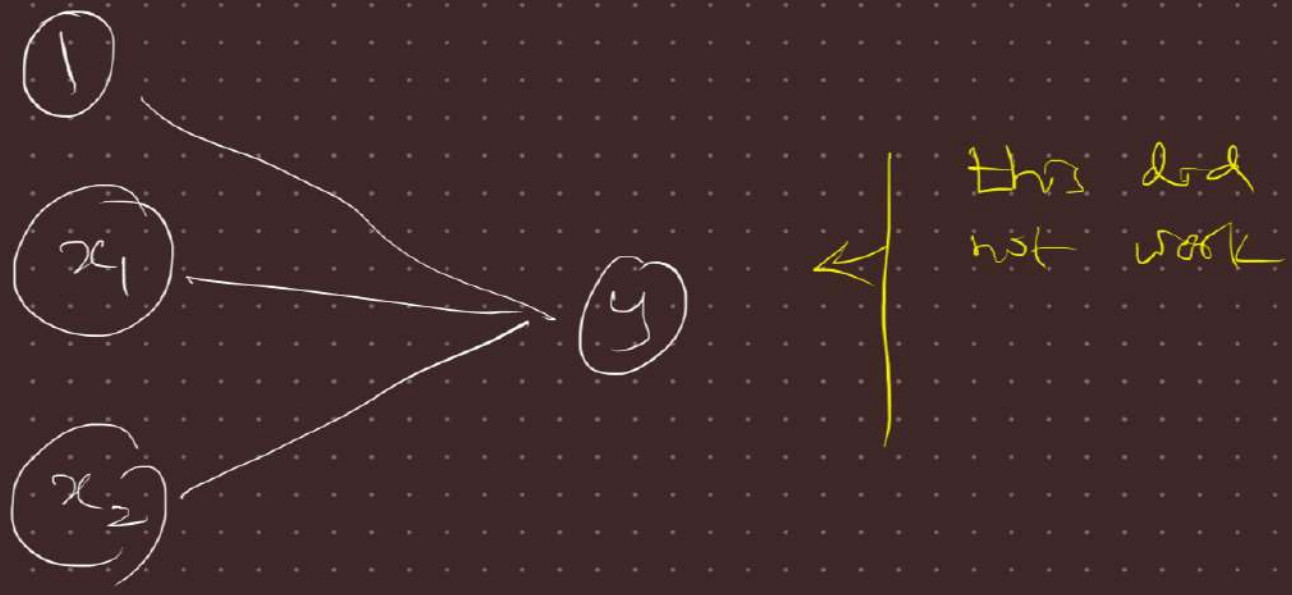
Intermediate output 2 (h_2)

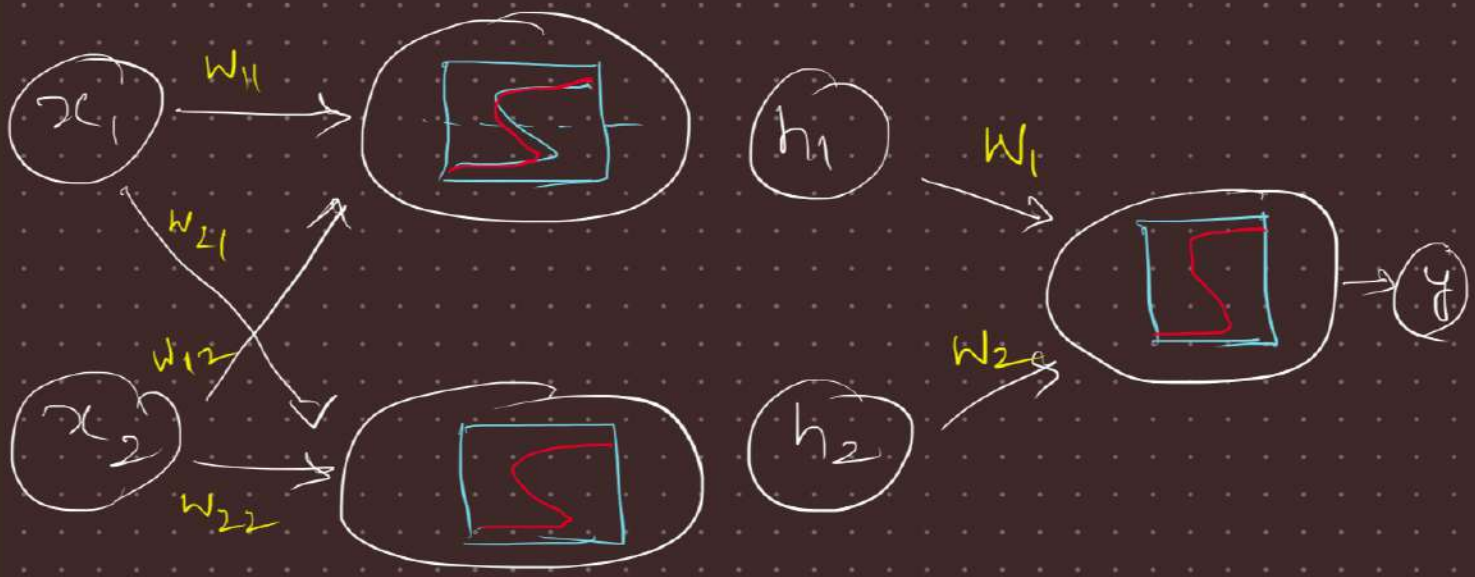
↓

predicted values from Model 2

Intermediate output h_1, h_2 Vs y







In the above example, instead of building one Logistic regression line, we are building three logistic regression lines.

NEURAL NETWORK INTUITION

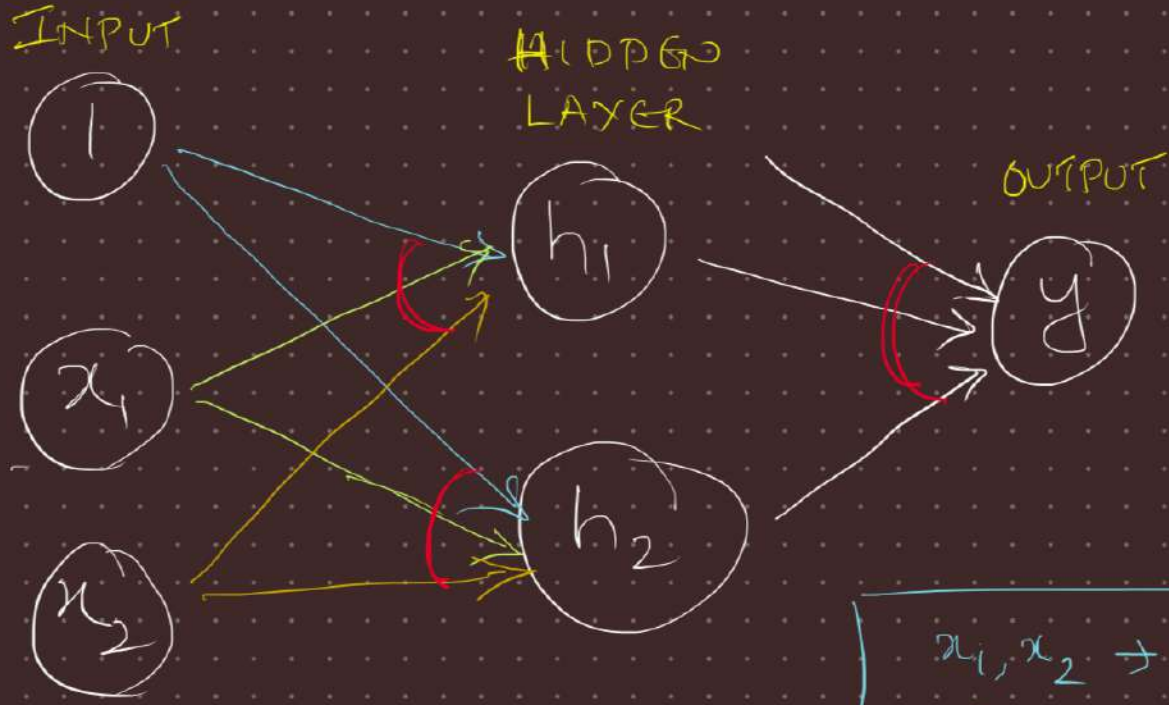
To predict y, we have used h_1, h_2

To predict h_1, h_2, we have used x_1, x_2

y = function of x_1, x_2

Neural Network will automatically build h_1, h_2 within one shot and give us the output (y)

NEURAL NETWORK AND VOCABULARY



Intermediate outputs are nothing but Hidden layers

x_1, x_2 → inputs

1 → bias term

W_i are weights

$1/(1+e^{-y})$ is the sigmoid function

y is output

In reality :

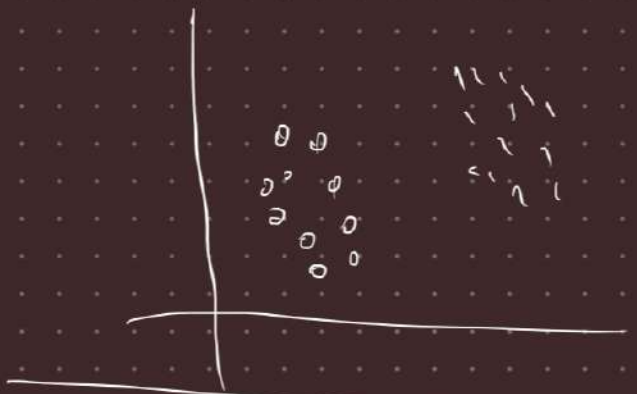
Intermediate outputs ⟹ Hidden Layer

Decision boundaries ⟹ hidden nodes

there is nothing like intermediate outputs or decision boundaries, instead we call them as Hidden layers and hidden nodes.

Note : Any complex data can be solved using ANN

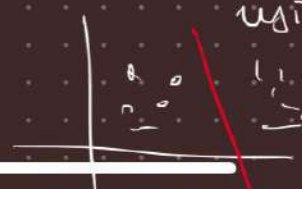
Test



→ To solve this problem how many hidden nodes are required?

Ans : Zero hidden nodes as it can be solved using Logistic Reg

Log. Reg is also a ANN but with zero hidden layer or node.



NEURAL NETWORK ALGORITHM

- Supply [x, y, #hidden nodes] for Neural Network Algorithm (iteratively)
- by trail & error of giving #hidden nodes, we will build the best model

example

Lets say Model M_1,
#hidden nodes reqd is & but we have used 16 } OVERFITTING

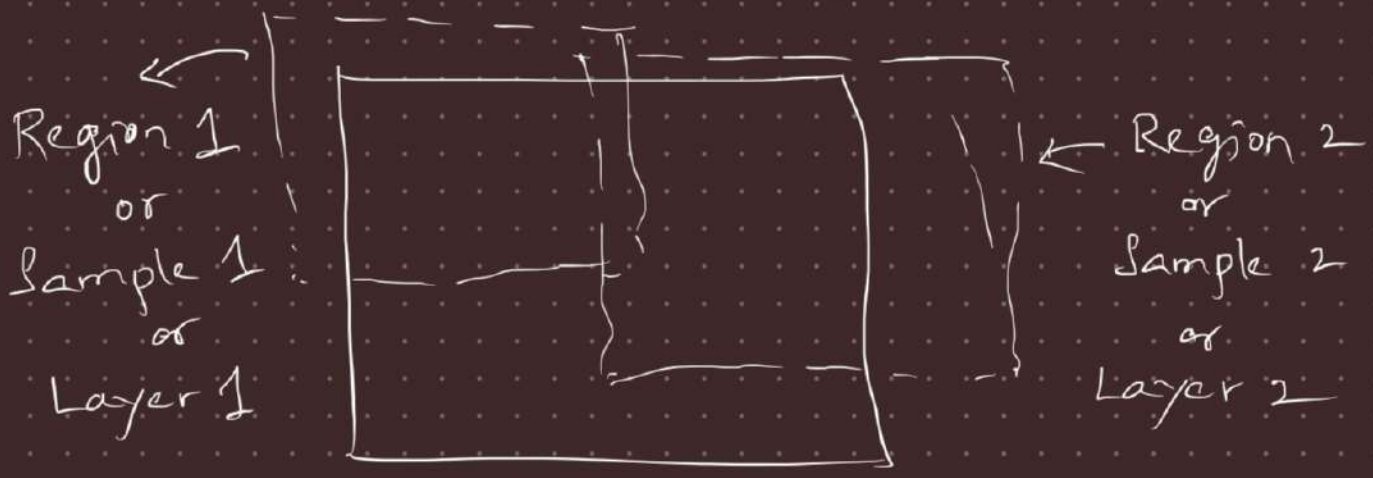
Lets say Model M_2,
#hidden nodes is & but we have used 2 } UNDERFITTING

M_1 — 16

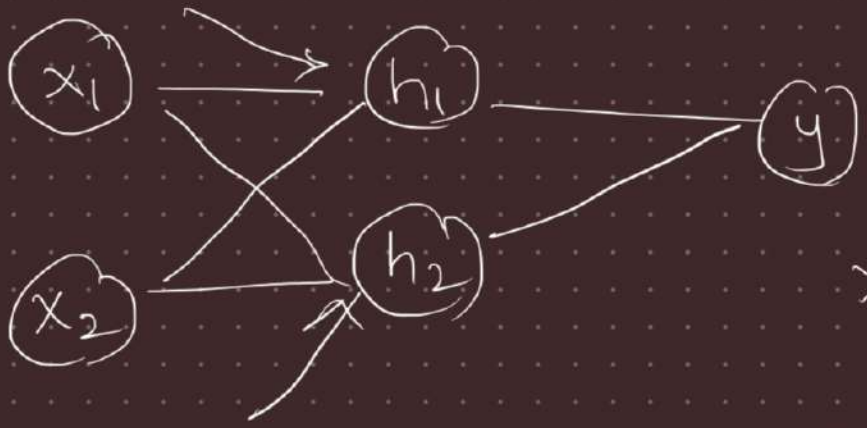
M_3 — ↓ you will somewhere in the middle

M_2 — 2

ANN → layered Logistic Reg Model



Understand Neural Network Algorithm



you have
x_1, x_2, y, # hidden nodes
you are asked to find weights

Weights are calculated using algorithm

Theorized → 5 steps

Neural Network Algorithm is all about finding weights

Step (1) : Randomly select some weights

↳ idea is start random, iteratively change them till you get optimal weights

Step (2) : Supply the training values (x_1, x_2) and perform the calculations forward.

↘

FEED FORWARD STEP

at the end of feed forward, you will get y_{pred} values

Step (3) : Calculate the error at the output. Use the output error to calculate error fractions at each hidden layer

↘

since you have provided random values, there will be error at each stage.

Back propogation

Calculate Error signals backward

what is this?
finding the error contribution at each layer.

Step (4) : Update the weights to reduce the error, recalculate and repeat the process.

BACK PROPOGATION ALGORITHM

In simple, all these steps happen in the backend we don't get to see this.