


Neural Network and Deep learning (supervised classification)

- 1 - take logistic regression for ex say we have a img and the pixel intensities 0-1 for ex (black to white) are features and we are trying to predict a target class (0-9) for the number given

Ex input:



num 1

pixel intensities (0-1) (black to white)

0	0	0	0	0	0	0
0	0	0	6	7	0	0
0	0	5	7	6	0	0
0	0	4	9	4	0	0
0	0	0	9	5	0	0
0	0	0	0	0	0	0

- this is difficult to do with Logistic Regression as the number 1 does not always look the same it slightly different each time

Ex

0	0	0	0	0	0
1	1	1	1	2	1
2	2	2	2	2	2
3	3	3	3	3	3
4	4	4	4	4	4
9	9	9	9	9	9

- 2 - But all the numbers have commonality like for ex 1 has a dominating vertical line a 9 has a vertical line + circle 0 does not have any lines etc.

7.2 Ex: Feature eng

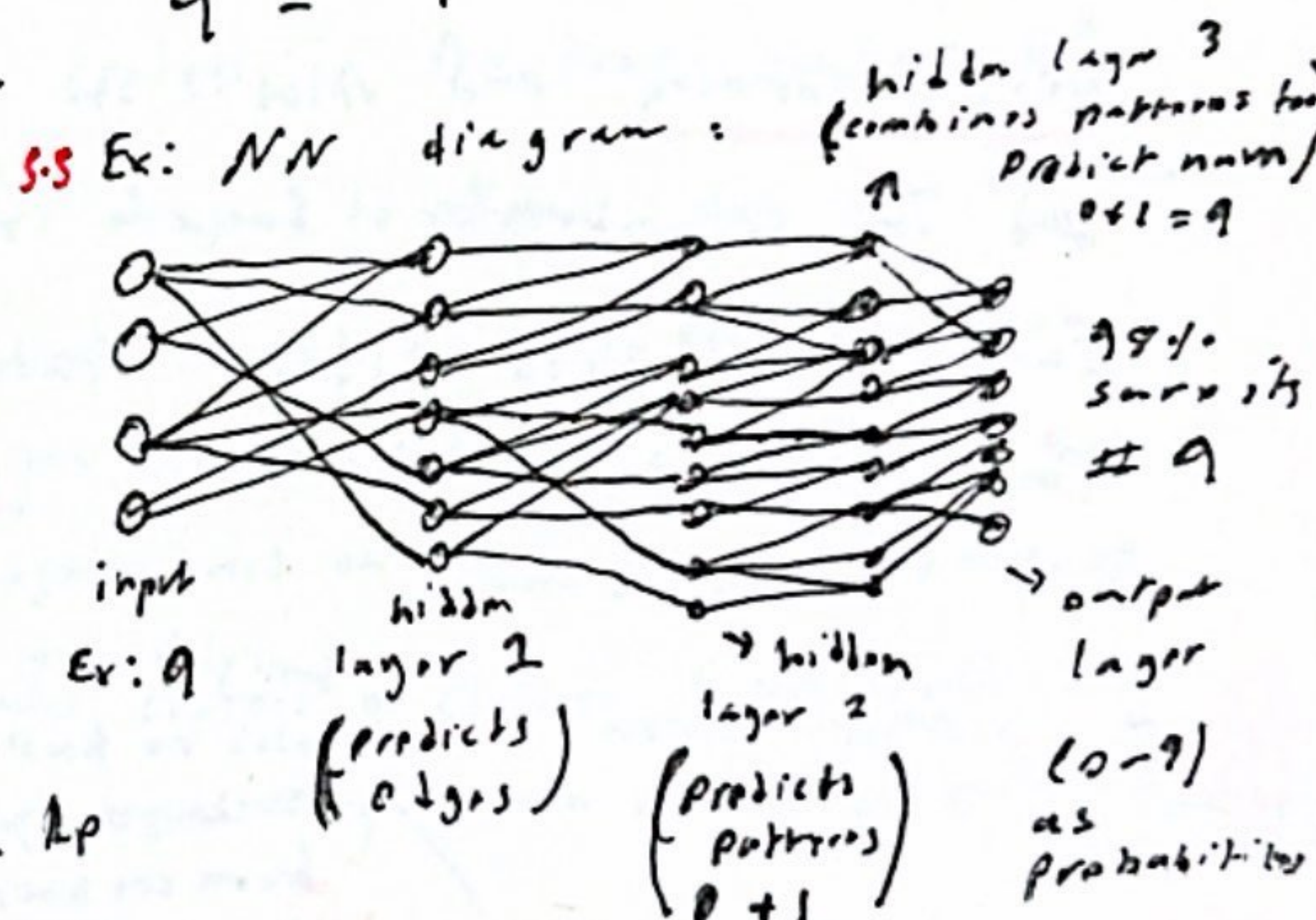
$$9 = 0 + 1$$

$$8 = 0 + 0$$

$$4 = 1 + 1 + -$$

the NN can find many more features this is a perfect case

- 3 - We can manually engineer these things and add them as features but NN like kernel function do this implicitly it automatically without any guidance. we do this by adding additional layers between the input and output layer these are unknown variables.



- 4 - in its simple form this is a single layer perceptron and the hidden layer now represents some unknown feature if in a perfect case it can be the stick line of number 1

- 5 - Now instead of predicting 1 directly we predict the hidden layer with input so predicting vertical line first then number after that (target)

don't predict 9 directly. predict what makes 9 first "0" "1"

HL 3 has no idea what a "0" is made from in what pixels make a "0" then HL 2 did that for us. this HL 3 only predicts what "0" and "1" make is 9

each hidden layer predicts another hidden layer, each layer the problem gets easier until we get to answer (only 9 0 neurons)