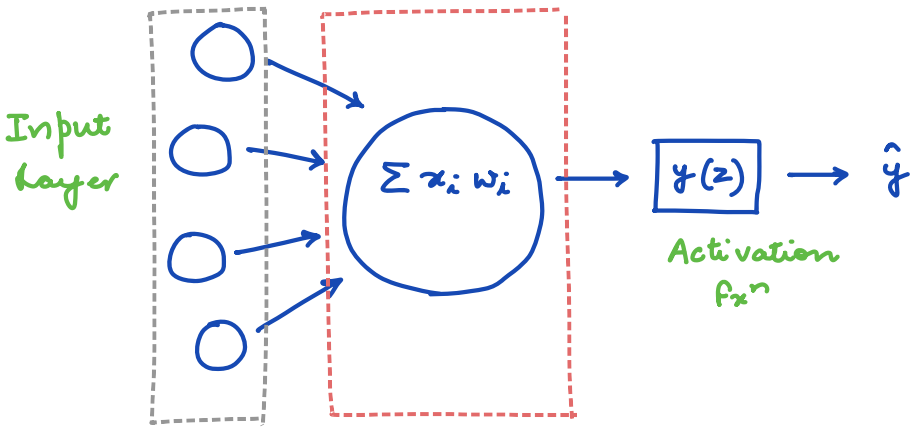
Learning Goals:

- Understand the Multilayer Perceptron (MLP) Architecture
- Understand Forward Propagation & Backpropagation, Loss & Activation functions
- Making Predictions
- Checking classification performance on different datasets
- Classification Project

Part 1 Neural Architecture

1 Layer Network

↳ only 1 node

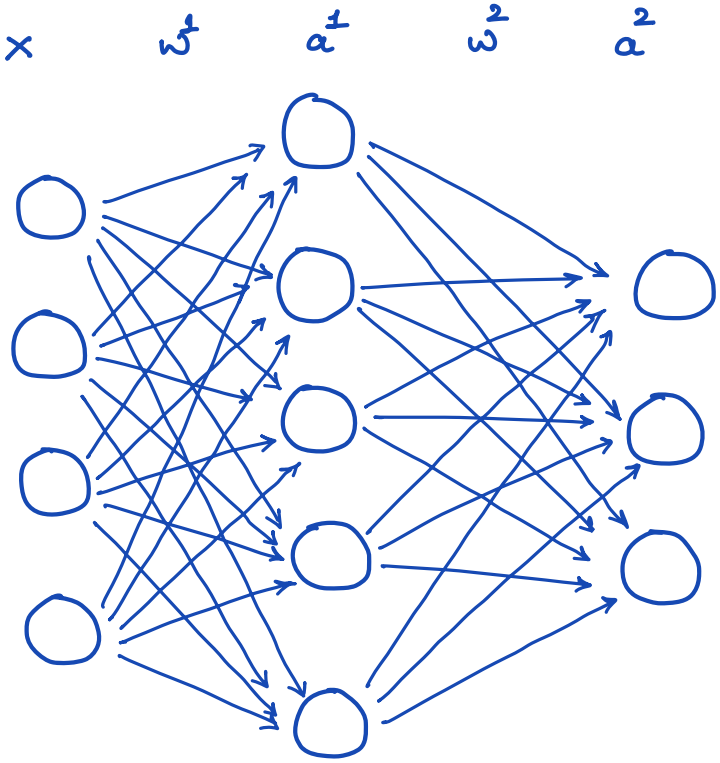


It is called 1 layer Network bcz it is a true layer. Input layer doesnot have any parameter.

- Simplest Unit Perceptron
- Only one output unit
- No hidden units
- Input is not counted in layers

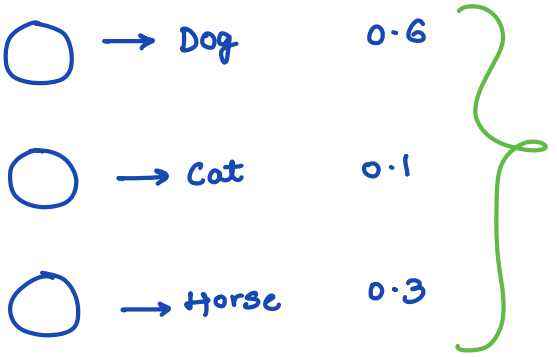
This type of network is not very powerful it works as linear separator only. It gives linear decision boundary.

2 Layer Network



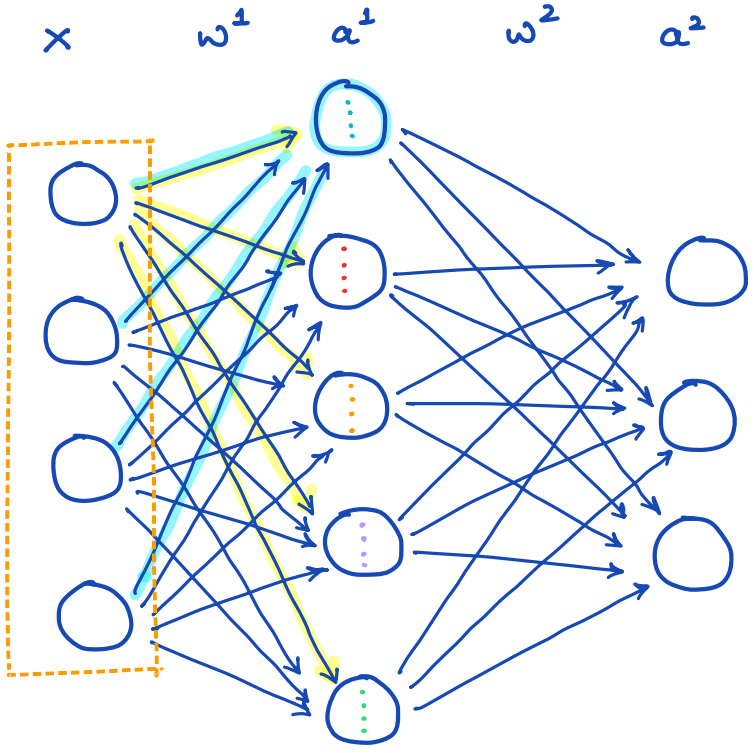
You can have any no. of units in output layer or in hidden layer. As you increase the no. of units the complexity of function also increases. You can represent more complex boundaries as you increase the no. of units.

No. of units in output layer depends on the no. of outputs that you want to have. Let us say you are doing a classification problem and you want to predict if given image is of a dog or cat or horse. In this case you are having 3 outputs and each output will represent the probability of dog, cat and horse.



class with maximum probability is your prediction. final output is going to be dog. Take argmax of probability

Input is not counted as a layer bcz there are no weight parameters that we are going to learn.

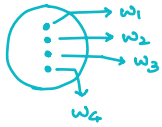


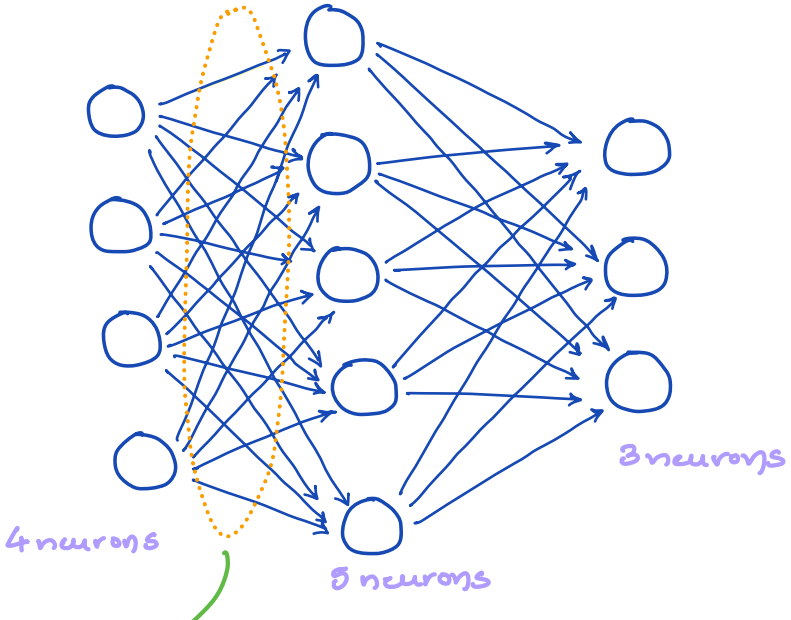
Layer 1
Input Layer

Layer 2
Hidden Layer

Layer 3
Output Layer

Every input in the 1st layer is connected with every other node in the 2nd layer.
With every edge there are going to be weights.
Every node is going to have 4 inputs.





You can represent all the parameters of this layer in form of matrix and it is called the weight matrix of the first layer.

$$W^{[1]} = \begin{bmatrix} \circ & \circ & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ \end{bmatrix}_{4 \times 5}$$

4 → no. of neurons in $(l-1)^{th}$ layer.

5 → no. of neurons in l^{th} layer.

Each column represents weights learned by a single neuron.

$$W^{[1]} = \begin{bmatrix} | & | & | & | & | \\ w_1^{[1]} & w_2^{[1]} & w_3^{[1]} & w_4^{[1]} & w_5^{[1]} \\ | & | & | & | & | \end{bmatrix}$$

weights associated with 1^{st} neuron of 1^{st} layer

weights associated with 3^{rd} neuron of 1^{st} layer.

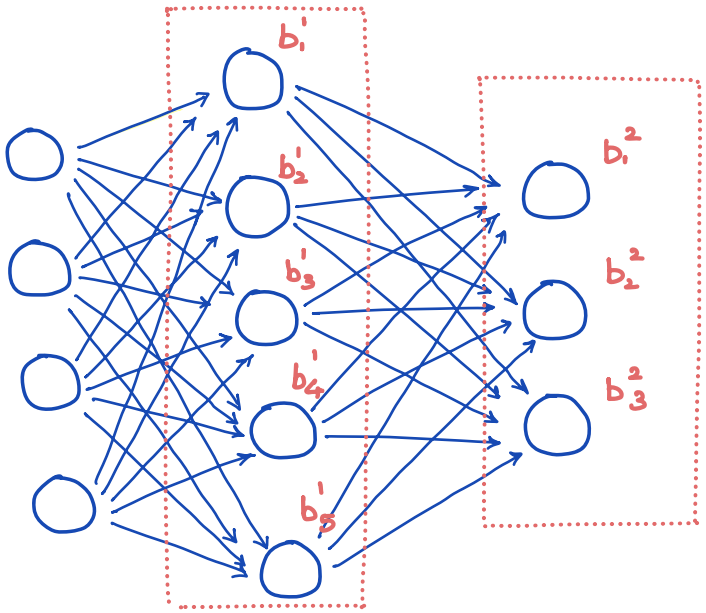
$w_i^{[l]}$ represents weights of i^{th} neuron in l^{th} layer.

$$W^{[2]} = \begin{bmatrix} & \\ & \\ & \\ & \\ & \end{bmatrix}_{5 \times 3}$$

Goal of neural network is to learn these parameters in

order to minimize the loss or to increase the probability of correct class.

Every neuron is going to have bias term.



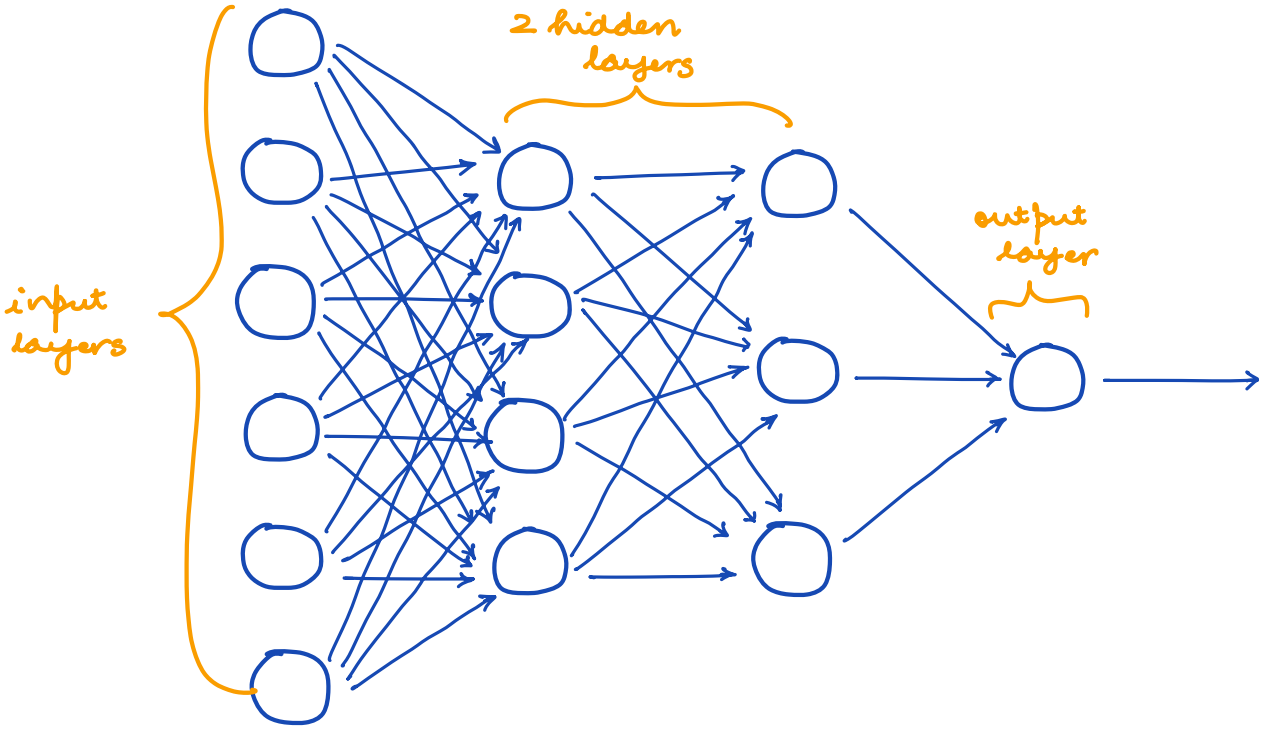
$$b^{[1]} = \begin{bmatrix} b_1^1 \\ b_2^1 \\ b_3^1 \\ b_4^1 \\ b_5^1 \end{bmatrix}_{5\times 1} \qquad b^{[2]} = \begin{bmatrix} b_1^2 \\ b_2^2 \\ b_3^2 \end{bmatrix}_{3\times 1}$$

No. of parameters your model is going to learn $= (4\times 5) + \underbrace{5}_{\text{bias}} + (5\times 3) + \underbrace{3}_{\text{bias}}$

- Two layers : 1 Hidden + 1 output
- Multi layer Perceptron or Feed forward Net or Deep forward Net.

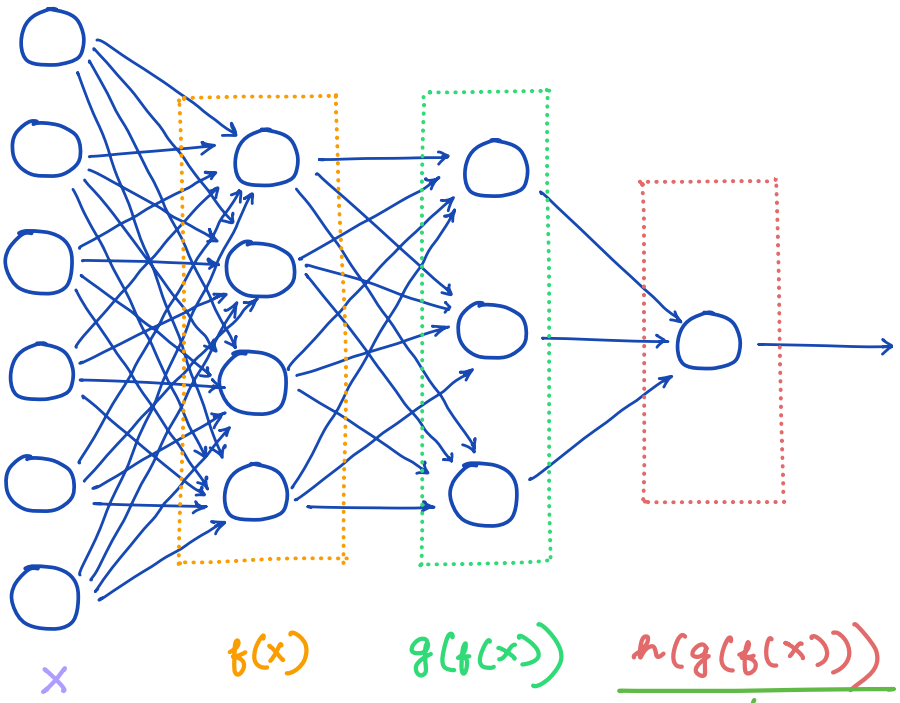
3 Layer Network

↳ we will increase the no. of hidden layers



As you increase the no. of hidden layers, network will become more and more deep (it can represent a complex fn")

- 3 Layers: 2 Hidden + 1 output



complex fx^n and combination of non linear fx^n.

↳ each neuron will apply activation fx^n and that activation fx^n is going to give a non linear fx^n. That fx^n can be a sigmoid fx^n or relu or tanh.

In $h(g(f(x)))$ you are combining many non linear fx^n. Our task in neural network is to learn complex fx^n's which are able to map your input x with desired output y.

$$f(x) \rightarrow y$$

our goal is to learn this fx^n f which maps input x to class y and does it with maximum accuracy.

If you have atleast 2 layers then this type of network is called multilayer perceptron (MLP) or feed forward net or deep forward net.