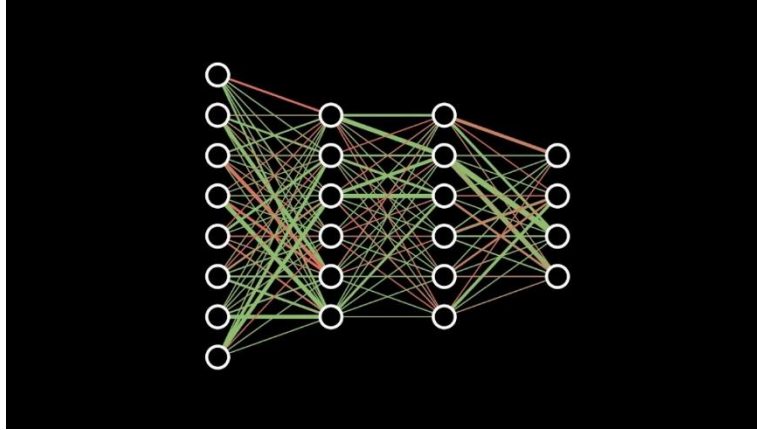


Simple Multi-layered Neural Network

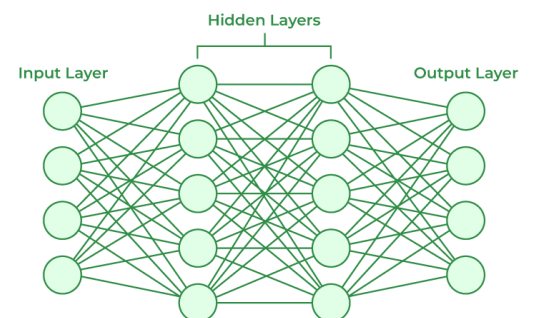


Network Structure

- Setup
- Feed Forward
- Back Propagation

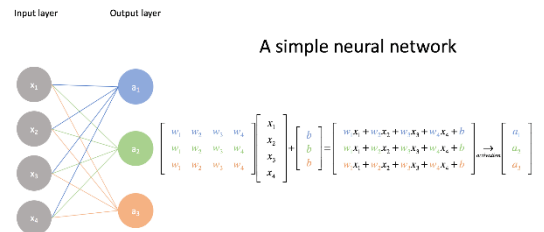
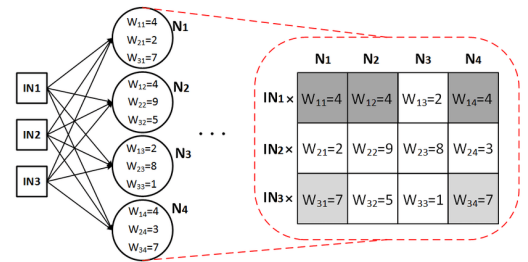
Setup:

- There is 1 input layer where you pick the amount of input nodes
- You can pick the amount of hidden layers and the amount of nodes all layers will have
- You choose any number of outputs that fit the problem your trying to tackle



Feed Forward:

- The feed forward algorithm starts with looking at the input layer.
- The inputs are whatever data you want the network to analyse.
- For each layer you have you also have a weight matrix.
- Each node in the next layer will be calculated by looking at the input matrix x the weight matrix.
- The network does this all the way to the output layer which is your result.



This alone is not enough to get any meaningful results. What you have to do now is “train” the network based on how wrong it was. That is what back propagation will do.

Back Propagation

- To teach a network anything you need to show it data, have it guess and then use the error margin.
The math here is pretty heavy so ill try to cover it:
- Give the network its guess and the correct answer in matrix form
- Start “tuning” the weight matrices.
This is done by using a method called gradient decent.
We compute how much we should tune the weights by using this formula:

$$\Delta W = lr * E * \delta' * O * I^T$$

lr = Learning Rate

W = Weight Matrix

E = Output Error Matrix

δ' = Sigmoid prime (Activation Function)

O = Output Layer

I^T = Input Layer

- You do that from the output layer and backwards until you reach the input layer
- This image here shows how a single “neuron’s” weights will be influenced by back propagation

