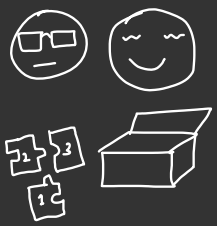
Neural Networks

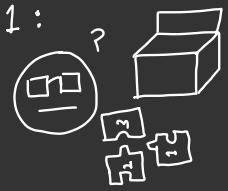
brain

connections



goal to put together the pieces of the puzzle

#step 1:



→ tries to find pattern by looking at the pieces (the input)

#step 2:

→ starts putting together the pieces (processing the data through layers of network)

#step 3:

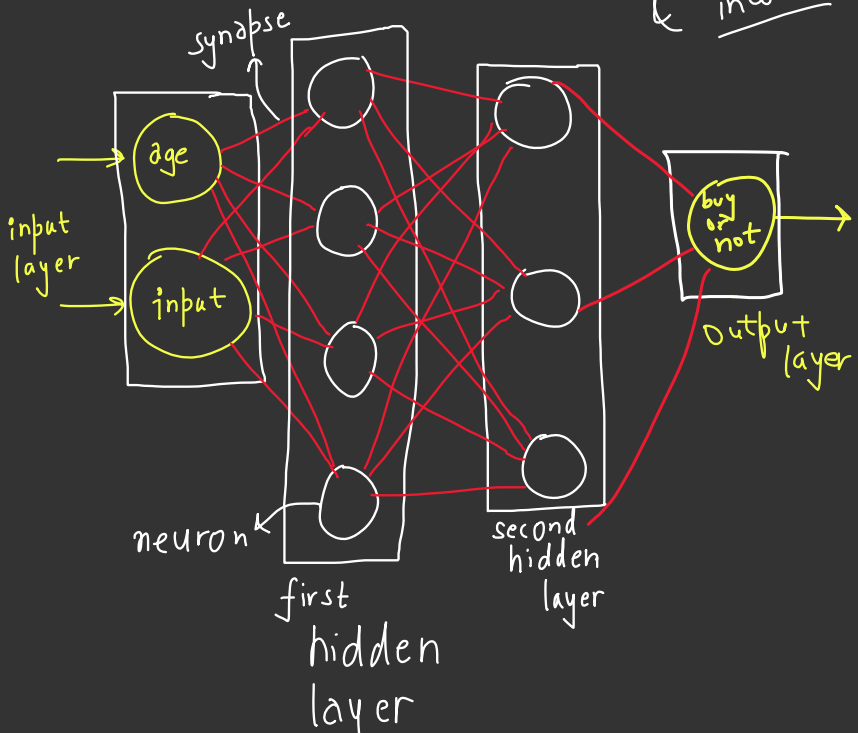
as more pieces come together the smart kid starts to see the bigger picture (make predictions based on processed data)

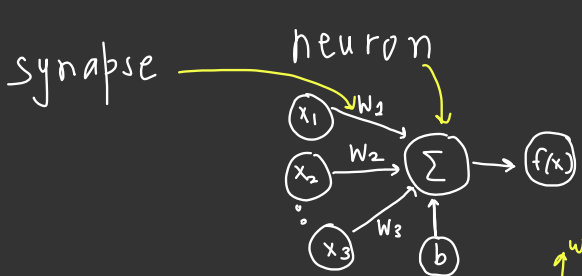
#step 4:

the kid adjusts how to put together the pieces based on whether they were right or wrong (learning from feedback)

Working of Neural Network

example, whether a person will buy a product based on their age & income





Output from one neuron

$$z^{(i)} = w_1 x_1^{(i)} + w_2 x_2^{(i)} + \ldots + w_k x_k^{(i)} + b$$

weights · inputs · bias

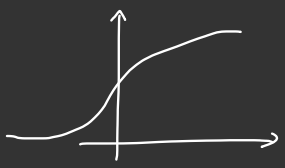
Activation Layers

Input Layer

$$\begin{bmatrix} x_1 & x_2 & \cdots & \cdots \\ \vdots & & & \\ & & & \end{bmatrix}$$

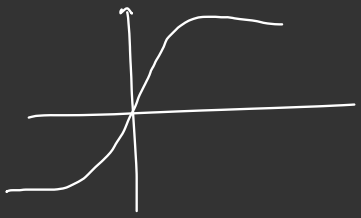
sigmoid function

$$S(x) = \frac{1}{1+e^{-x}}$$



hyberbolic tangent function

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$



etc.

Why use Activation Layer?

- To introduce non-Linearity in the data.
- To decide which neuron to activate by how much.
- Maps data to a known range to stabilize training
- Without Activation Layer everything is just Linear sum & multiplication, & we wont get interesting output from the Neural Network.

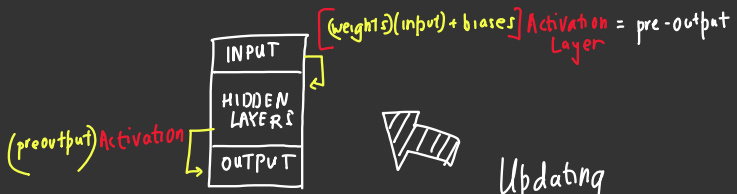
Softmax Activation Fun°

output layer $\longrightarrow \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \longrightarrow$ probabilities

Training & Feedback

tweaking weights & biases according to the situation it should be trained against

taking pre output situations & running it through the NN then balancing the weights & biases to get output closer to the real result



Updating the system weights & biases to better suit the program

Backend Training & Optimisation