We can use a neural network to predict demand of a particular article of clothing.

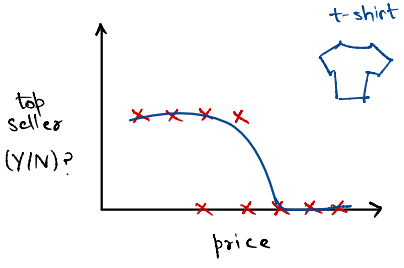
Suppose

$$x = \text{price} \quad \text{and} \quad a = f(x) = \frac{1}{1 + e^{-(\vec{w} \cdot \vec{x} + b)}}$$

(x: input; a: output)

"activation"

- this term was taken from neuroscience and refers to how much a neuron is sending a high output to other neurons.



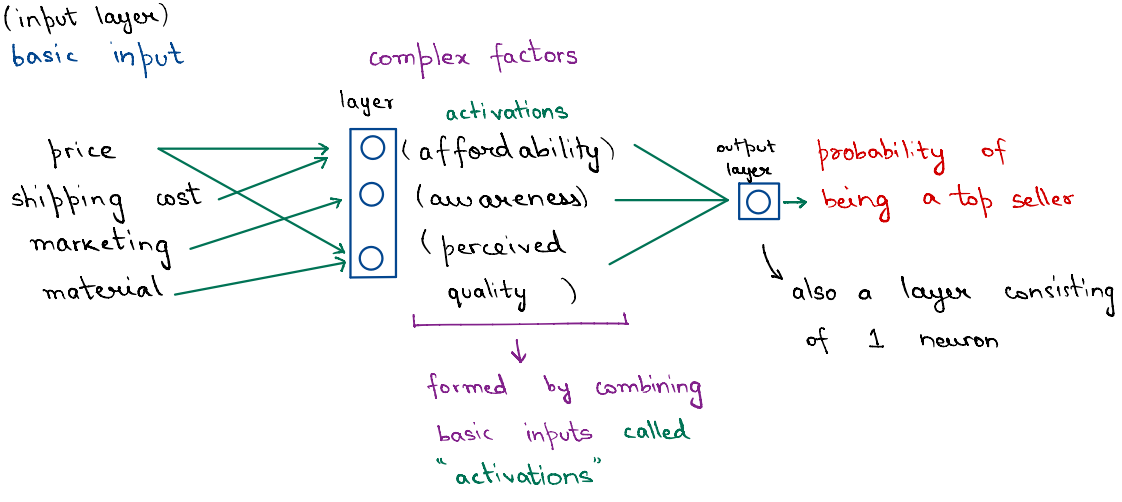
$x \rightarrow$ [neuron] $\xrightarrow{a}$ probability of this t-shirt being a top seller

basically a small computer that takes input and outputs something.

building a neural network just means taking a bunch of these computers and grouping them

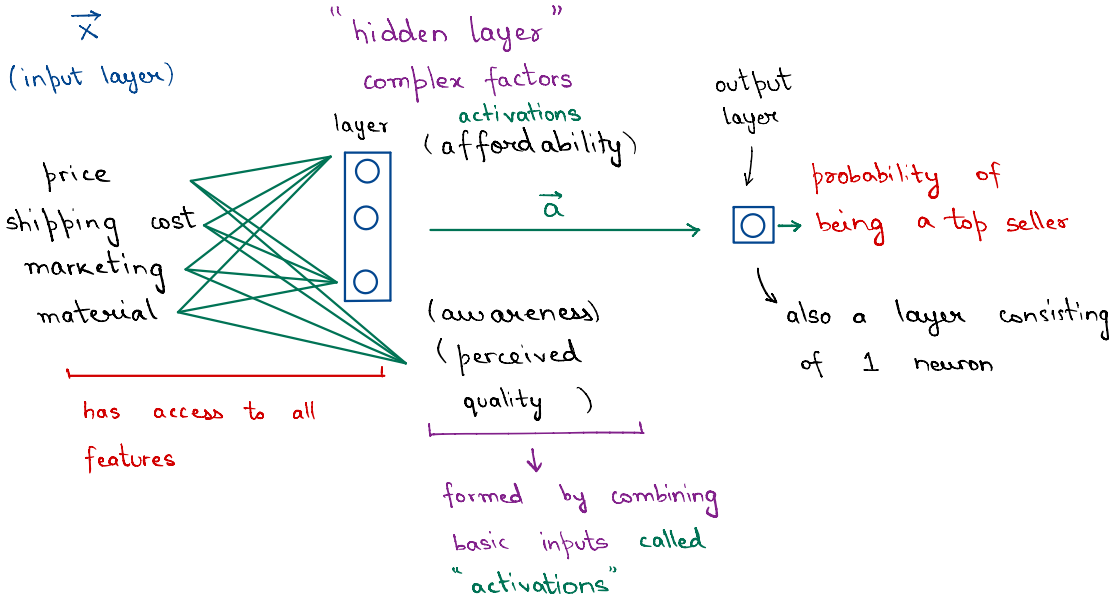
In this case we will calculate demand using more complex factors.

These complex factors are created from basic inputs using a technique called feature engineering.



In Summary,

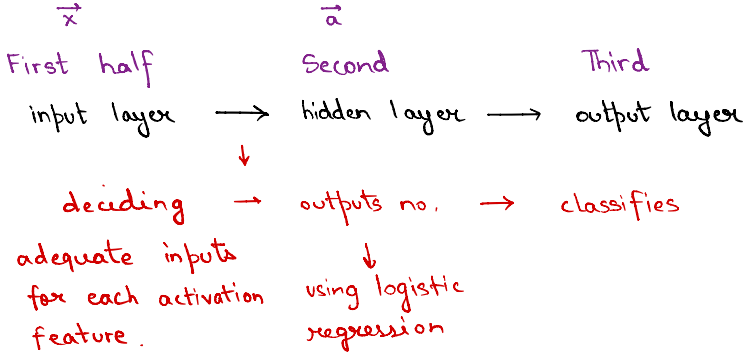
input is 4 nos. → 3 nos. are activations → 1 no. is input



- Two vector layers are basically there, $\vec{x}$ and $\vec{a}$.

- We can't always decide which feature will be grouped together, so we connect every input and let the model decide useful inputs for the activations by setting correct parameters.

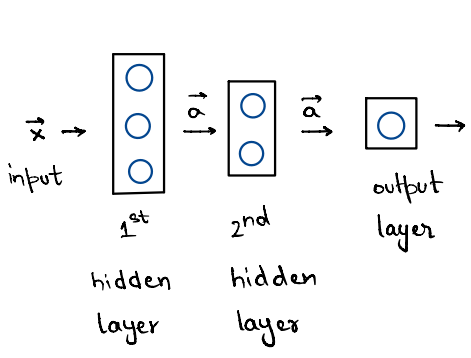
If you think about it,

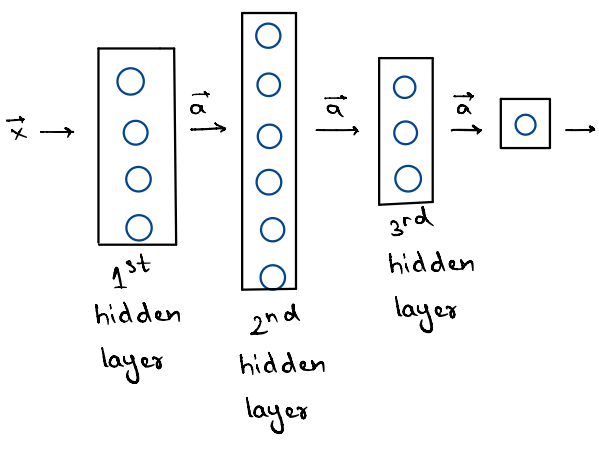


Basically, the above example of neural network is just logistic regression except that the features are not already given and the algorithm learns its own features using the inputs given.

In the hidden layer we had 3 features - affordability, awareness and perceived quality, but usually there are no features pre-given, instead the algorithm learns on its own about what inputs can be used to feature engineer new features for the hidden layer.

Multiple hidden layers
(called multilayer perceptron)





Each neural network has its own architecture based on its need. It decides the no. of hidden layers and no. of neurons each layer should have.