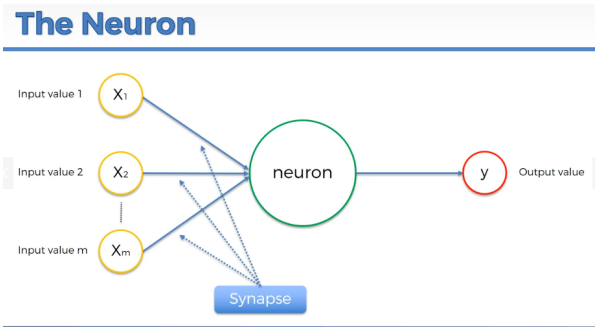
Introduction to ANN

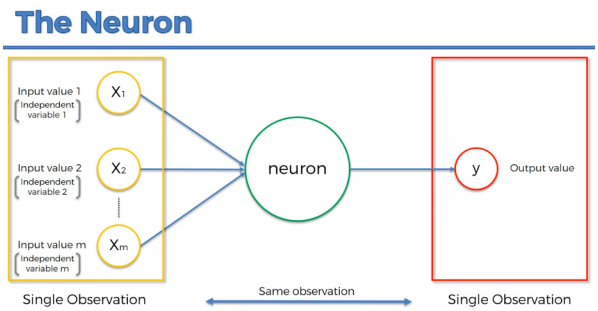
https://www.superdatascience.com/blogs/the-ultimate-guide-to-artificial-neural-networks-ann

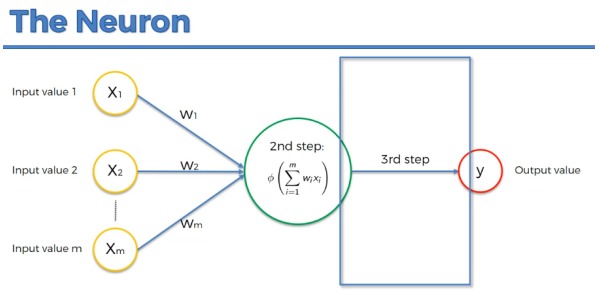
- The brain cells, neurons, are consists of three parts:
  - The body of the neuron
  - Dendrites - receiver of the signal for the neuron
  - Axon - transmitter of the signal for the neuron

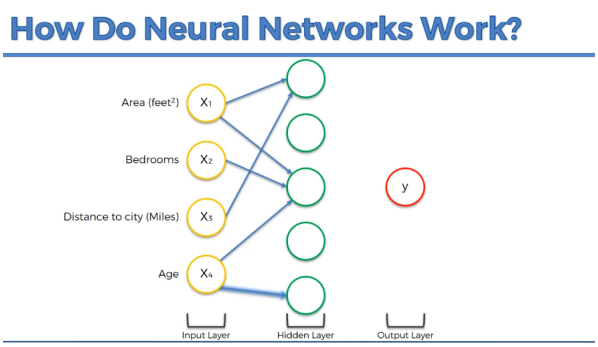
Dendrites of a neuron connects with axon of another neuron. However, there is no physical connection between them. The gap is called `synapse`. The connections in an ANN comes from the idea of synapses. Because calling the connections dendrites of axon will bring the confusion of which artificial neuron connected to which artificial neuron.

- Neuron in machine language also sometimes called Node.
- A Neuron is connected on the left with the input neurons and gets input values. On the other hand, it is connected with the output neuron on the right and transmits the output value.
- Input layer is similar to human senses. All independent variables (input values) corresponds to a row in the dataset. The input variables have to be **standardized** or **normalized** before fed to the input neurons.
- It is weights that make possible the artificial neural networks to learn. It learns which input values are important and which are not by optimizing the weights.
- Activation function decides if the neuron will pass the signal or not.
- In the most basic form of an ANN, there is no hidden layer between input layer and output layer. There are only synapses and weights. However, the power of ANN comes from adding hidden layers.









Steps in Training ANN

1. Randomly initialise the weights to small numbers close to 0 (but not 0).
2. Input the first observation of your dataset in the input layer, each feature in one input node.
3. Forward-Propagation: from left to right, the neurons are activated in a way that the impact of each neuron's activation is limited by the weights. Propagate the activations until getting the predicted result $\hat{y}$.

4. Compare the predicted result, $\hat{y}$, to the actual result, y. Measure the generated error, e.
5. Back-Propagation: from right to left, the error, e, is back-propagated. Update the weights according to how much they are responsible for the error. The learning rate decides by how much the weights will be updated.
6. Repeat steps 1-5 and update the weights after each observation (Reinforcement Learning). / Repeat steps 1-5 but update the weights only after a batch of observations (Batch Learning).
7. When the whole training set passed through the ANN, that makes an *epoch*.
8. Do more epochs.

Resources:

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/7b11e353-fe00-45bd-93e4-fa6eb72b6764/lecun-98b.pdf