# Neural Networks

* Neural Networks are a way of modeling biological neuron systems mathematically.
* These networks can then be used to solve tasks that many other types of algorithms can not (e.g. image classification)
* Deep Learning simply refers to neural networks with more than one hidden layer.

# ML is mainly classified (there are other classifications as well):

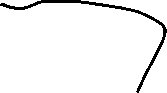
## Supervised learning (most used):

* (Informal definition): **right answers**(labels) are already given, and the machine has to predict more right answers
* Supervised learning problems are categorized into "**regression**" and "**classification**" problems
* Commonly used in applications where we use historical data to predicts likely future events
* Model (or network) receives a set of inputs and corresponding correct ouputs, model compares its predicted output with correct output to find errors and modify model accordingly
* Regression will deal with **continuous data** or **real valued output** (because it has to predict the output value which is not classified in to set of categories like **classification**), classification will deal with discrete data

Example:



* Given data about the size of houses on the real estate market, try to predict their price. Price as a function of size is a continuous output, so this is a regression problem.



* **Classification** will deal with discrete output.
* We could turn the above example into a classification problem by instead making our output about whether the house "sells for more or less than the asking price." Here we are classifying the houses based on price into two discrete categories.

### ML process for supervised learning

Diagram

Description automatically generated

* We acquire data from multiple sources, like censors, physical records …
* We remove unwanted data and format data as per our needs using pandas in python and python
* We split data in to 2 parts
  + Train data: 70 % of original data
  + Test data: 30% of original data
* We use train data to train model and test data to test it
* If we are not satisfied with test results then we can go back to model training and building, then start the process again from there
* If we are satisfied with test results, we can go ahead and deploy the model for realtime predictions

#### Important Note

Just doing a single split will split the data in to train and test, when you test model, you get an accuracy and need less to say that is how the model is going to perform in real time. So it will be really better if we can split the data in to 3 steps

1. train the model,
2. validate the model,
   1. if the accuracy is good proceed to step 3
   2. If the accuracy is not good, then adjust the parameters and try validating the model again (continue the process from step 2 after adjusting parameters)
3. Test the model after setting hyper parameters

## Unsupervised Learning:

* A approach where we have the **same label** or **no label** as output, clustering comes under unsupervised category.
* Unsupervised learning allows us to approach problems with little or no idea what our results should look like. We can derive structure from data where we don't necessarily know the effect of the variables.
* We can derive this structure by clustering the data based on relationships among the variables in the data.
* With unsupervised learning there is no feedback based on the prediction results.
* **Example:**
* **Clustering**: Take a collection of 1,000,000 different genes, and find a way to automatically group these genes into groups that are somehow similar or related by different variables, such as lifespan, location, roles, and so on.
* **Non-clustering**: The "Cocktail Party Algorithm", allows you to find structure in a chaotic environment. (i.e. identifying individual voices and music from a mesh of sounds at a [cocktail party](https://en.wikipedia.org/wiki/Cocktail_party_effect)).
* **Difference between clustering and non**-**clustering algorithms** : In clustering problems, the algorithms do not transform or change the data, it just groups them according to some characteristics. However, in non-clustering problems, the algorithm is transforming the data/input
* In a regression problem, our goal is to predict a continuous valued output
* Ex: time taken by an athlete to complete marathon, when we have some features such as age, length of marathon, previous track record
* A classification problem will have a discrete valued output of 0 or 1,sometimes there are more than 2 possible outputs
* Ex: if you are classifying the price of house, you can classify them in to, extremely priced (as 0), moderately priced (as 1), affordable(as 2)