



# *Hands on Lab on Neural Networks*

# AGENDA OF THE VIDEO

- What is neural network?
- ANN vs BNN
- Structure of a Single Layered NN
- Structure of a Double Layered NN
- Activation Functions in Neural Network
- Forward Propagation vs Backward Propagation
- How does NN differs from Perceptron?
- Pseudo Code for Coding Multilayer Perceptron in Sklearn
- Some important parameters of MLP
- Multilayer Perceptron in Sklearn for Classification
- Multilayer Perceptron in Sklearn for Regression
- NO GPU SUPPORT FOR SKLEARN NEURAL NETWORK
- Optimizing our model for best performance
- Task for you.....

# *What is Neural Networks?*

A neural network is a network of neurons (either Biological or Artificial) which helps humans to perform some complex task (BNN) or Computer to solve AI problems (ANN).

They are one of the most earliest developed algorithms. The reason we are seeing such a great interest in these nowadays is because of the greater amount of data as well as greater computational resource availability.

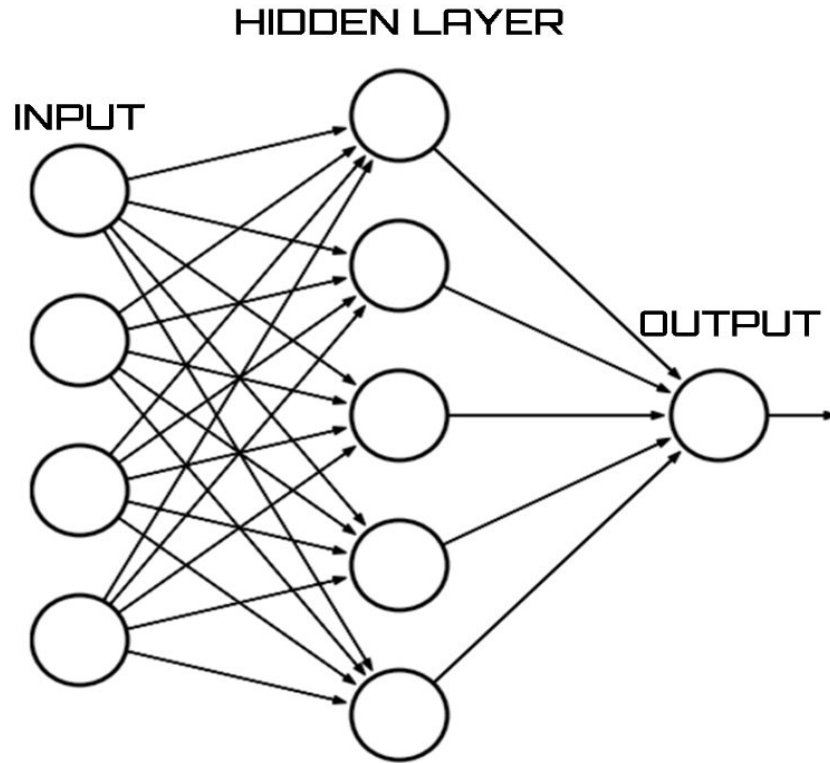
Both of them works on a principle whether **neurons fire or not**.

# *ANN vs BNN*

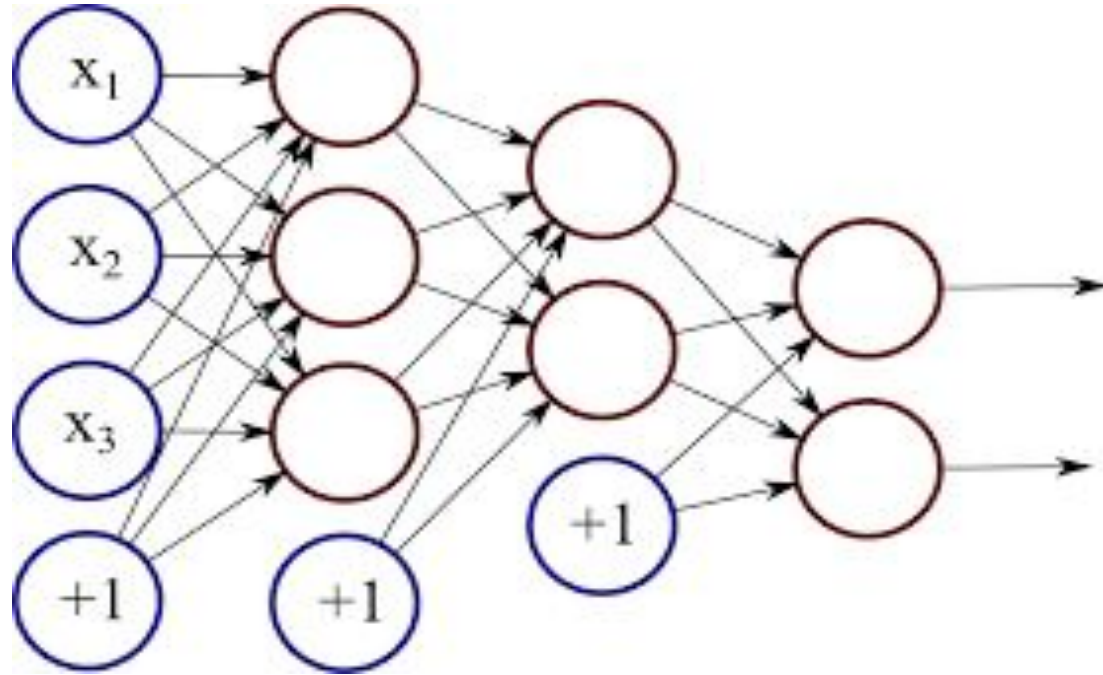
ANN stands for Artificial Neural Network which are made to solve AI tasks by humans.

BNN stands for Biological Neural Network which every human is having in their brains.

# Single Layered NN



# Double Layered NN

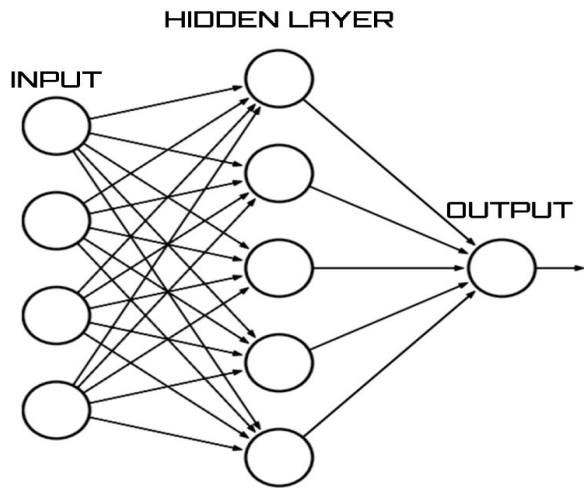


# Activations Functions in NN

Activation Function are those functions in NN which decides whether a Neuron is going to **fire or not**.

To decide that, we have activation function in NN but this happens almost automatically for BNN and we are not aware of which Activations Functions used by our brain to come up with the decision of firing or not firing of a neuron.

# Forward Propagation and Backward Propagation



Forward Propagation is term used for passing the data and weights in a forward direction all the way to the node or layer.

Once we get the output from our NN, then what we do is we run a backward propagation which takes that output and takes the actual output and then adjusts the weights associated with each of the neurons associated according to the error.



# *How does NN differs from Perceptron?*

Perceptron is also a very simple example of NN with no hidden layers and no activations functions.

Usually, in perceptron no hidden layers are involved and there is no concept of Activation Functions in Perceptron. So, a new version of Perceptron is called multi-layers perceptron has been developed and we are going to coding that in this session.

# Pseudocode for coding MLP (Multi-layer Perceptron)

```
#STEP. 1 Importing the necessary modules like sklearn's MLPClassifier, \
# MLPRegressor
import pandas,MLPClassifier, MLPRegressor

#STEP. 2 Import the data
data = # using pandas import the data

#STEP. 3 Make an instance of the above Class
clf = MLPClassifier()
reg = MLPRegressor()

#STEP. 4 Call its fit Method
clf.fit(X,y)
reg.fit(X,y)

#STEP. 5 Call the predict method to get the predictions
pred = clf.predict(X)
pred = reg.predict(X)
#STEP. 6 Using the actual labels and the prediction, check model performance
accuracy = sum(pred == y)/len(y)

#STEP. 7 Repeat the steps from 3 iteratively to come up with best models and
# Improve the accuracy of the model.
# Change some hyperparameters and re fit the above model to kind of see how does
# new configurations work.
```

# *MLP for Classification*

# *MLP for Regression*

# *Optimization of Performance of Neural Network*

# *NO GPU SUPPORT*

If you are interested in working with Neural Networks on your own projects, then I feel you should know that there is no GPU support for training Neural Networks for Sklearn.

This is why, it is not advisable to run Neural Network on really large datasets because it might take forever to train them.

For those projects, you might have to read about Tensorflow and keras.

## Task For you..

1. Fit a Multilayer Perceptron Algorithm on the Newsgroup data set which comes with the sklearn Library.
2. Optimize its performance by tweaking various parameters like you can try changing activation, number of hidden layers, number of neurons in each hidden layers etc.
3. See, if you can figure out what other Parameters are in the `MLPClassifier()` class in sklearn.