EX.NO : **DATE :**

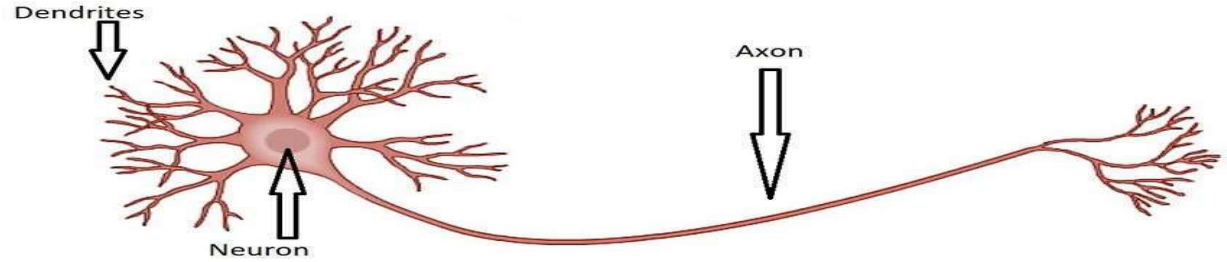
IMPLEMENTING ARTIFICIAL NEURAL NETWORKS FOR AN APPLICATION USING PYTHON - CLASSIFICATION

AIM :

To implementing artificial neural networks for an application in classification using python.

What is an Artificial Neural Network?

Artificial Neural Network is much similar to the human brain. The human Brain consist of **neurons**. These neurons are connected. In the human brain, neuron looks something like this…



As you can see in this image, There are **neurons, Dendrites, and axons.**

What do you think?

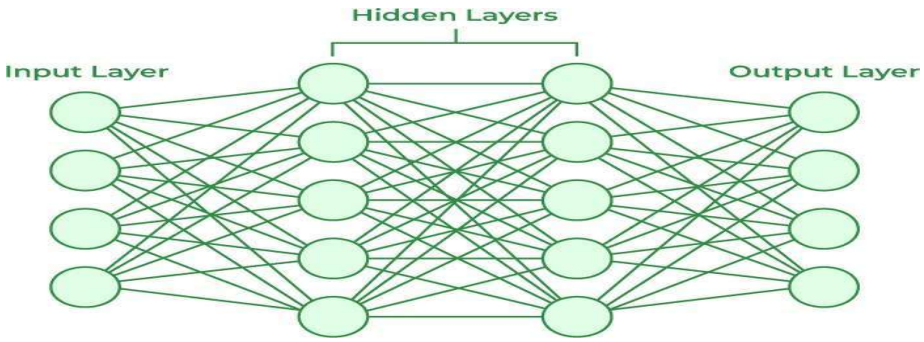
When you touch the hot surface, how you suddenly remove your hand?. This is the procedure that happens inside you.When you touch some hot surface. Then automatically your skin sends a signal to the neuron. And then the neuron takes a decision, **"Remove your hand"**. So that's all about the **Human Brain.** In the same way, **Artificial Neural Network works.**

Artificial Neural Networks

Artificial Neural Networks contain artificial neurons which are called **units**. These units are arranged in a series of layers that together constitute the whole Artificial Neural Network in a system. A layer can have only a dozen units or millions of units as this depends on how the complex neural networks will be required to learn the hidden patterns in the dataset. Commonly, Artificial Neural Network has an input layer, an output layer as well as hidden layers. The input layer receives data from the outside world which the neural network needs to analyze or learn about. Then this data passes through one or multiple hidden layers that transform the input into data that is valuable for the output layer. Finally, the output layer provides an output in the form of a response of the Artificial Neural Networks to input data provided.

The structures and operations of human neurons serve as the basis for artificial neural networks. It is also known as neural networks or neural nets. The input layer of an artificial neural network is the first layer, and it receives input from external sources and releases it to the hidden layer, which is the second layer. In the hidden layer, each neuron receives input from the previous layer neurons, computes the weighted sum, and

sends it to the neurons in the next layer. These connections are weighted means effects of the inputs from the previous layer are optimized more or less by assigning different-different weights to each input and it is adjusted during the training process by optimizing these weights for improved model performance.

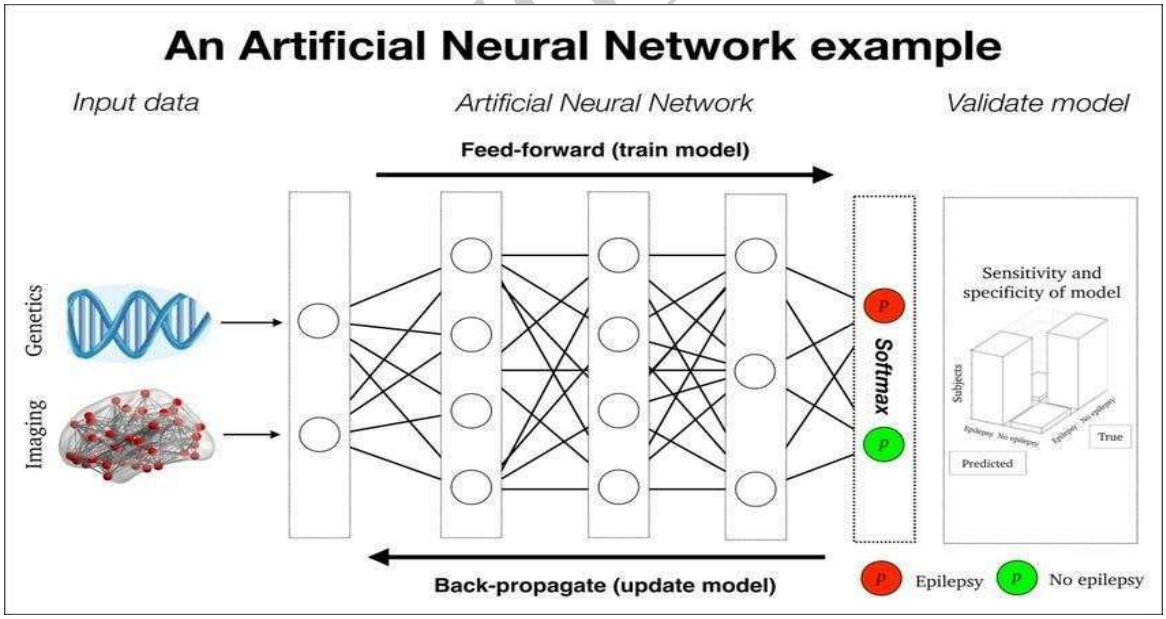


What are the types of Artificial Neural Networks?

1. Feedforward Neural Network
2. Convolutional Neural Network
3. Modular Neural Network
4. Radial basis function Neural Network
5. Recurrent Neural Network

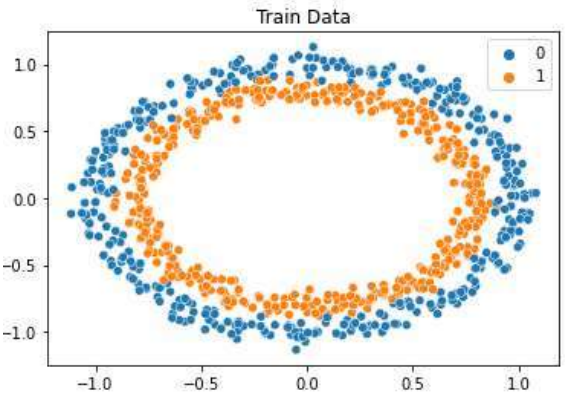
Applications of Artificial Neural Networks

1. Social Media
2. Marketing and Sales
3. Healthcare
4. Personal Assistants



SOURCE CODE:

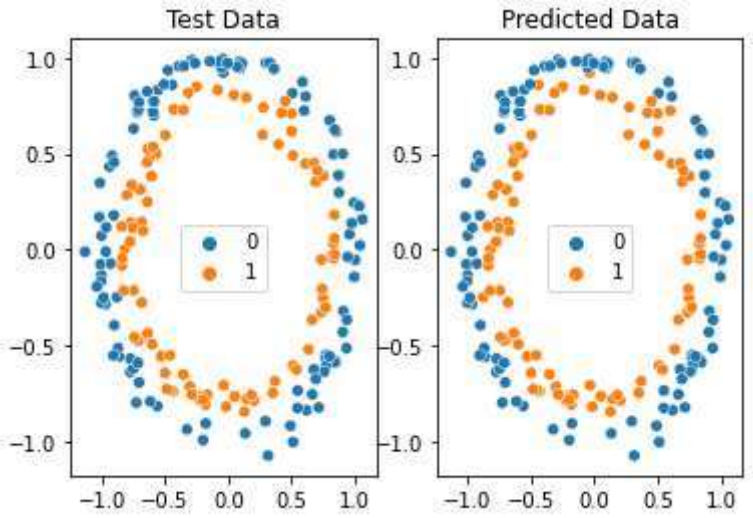
```
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import train_test_split
from sklearn.datasets import make_circles
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

X, y = make_circles(n_samples=1000, noise=0.05)

ns.scatterplot(X_train[:,0], X_train[:,1], hue=y_train)
plt.title("Train Data")
plt.show()
```




```
clf = MLPClassifier(max_iter=1000)
clf.fit(X_train, y_train)
print(f"R2 Score for Training Data = {clf.score(X_train, y_train)}")

print(f"R2 Score for Test Data = {clf.score(X_test, y_test)}")

y_pred = clf.predict(X_test)

fig, ax =plt.subplots(1,2)
sns.scatterplot(X_test[:,0], X_test[:,1], hue=y_pred, ax=ax[0])
ax[1].title.set_text("Predicted Data")
sns.scatterplot(X_test[:,0], X_test[:,1], hue=y_test, ax=ax[1])
ax[0].title.set_text("Test Data")
plt.show()
```

220701202

OUTPUT :



220701202

RESULT:

Thus, the code has been successfully executed, and the output has been verified successfully.