* **Feed Forward Backpropagation Neural Network:-**

**Code**:-

import tensorflow as tf

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

# Load your dataset or replace this with your dataset loading code

# For demonstration, let's use the famous Iris dataset from TensorFlow

from sklearn.datasets import load\_iris

iris = load\_iris()

X, y = iris.data, iris.target

# Convert labels to binary classification (0 or 1)

y\_binary = (y == 0).astype(int)  # Change the class based on your task

# Split the dataset into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y\_binary, test\_size=0.2, random\_state=42)

# Build a simple Feed Forward Neural Network

model = tf.keras.Sequential([

    tf.keras.layers.Dense(8, input\_dim=X\_train.shape[1], activation='relu'),

    tf.keras.layers.Dense(1, activation='sigmoid')  # Adjust the number of units based on your task

])

# Compile the model

model.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy'])

# Train the model

model.fit(X\_train, y\_train, epochs=50, batch\_size=32, validation\_split=0.2, verbose=0)

# Evaluate the model on the test data

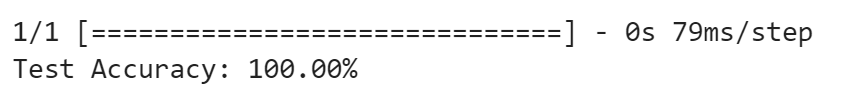
y\_pred = model.predict(X\_test)

y\_pred\_classes = (y\_pred > 0.5).astype(int)

accuracy = accuracy\_score(y\_test, y\_pred\_classes)

print(f'Test Accuracy: {accuracy:.2%}')

**Output:-**

****