16. Write the python program to implement Feed forward neural Network.

**AIM :** Feed forward neural Network.

**ALGORITHM :**

1. We import necessary modules: ‘numpy’ for numerical operations, ‘tensorflow’ for creating neural networks, and ‘sklearn’ for dataset loading, preprocessing, and splitting.
2. We load the Iris dataset and split it into features (‘X’) and target labels (‘Y’).
3. We one-hot encode the target labels using the ‘OneHotEncoder’.
4. We split the dataset into training and testing sets using ‘train\_test\_split’.
5. We create a simple Feedforward Neural Network model using the ‘Sequential’ API in TensorFlow. It consists of an input layer, a hidden layer with ReLU activation, and an output layer with softmax activation.
6. We compile the model by specifying the optimizer and loss function.
7. e train the model on the training data using the ‘fit’ method.
8. e evaluate the model on the test data using the ‘evaluate’ method and print the test loss and accuracy.

**PROGRAM :**

import numpy as np

import tensorflow as tf

from sklearn.datasets import load\_iris

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

from sklearn.preprocessing import OneHotEncoder

iris = load\_iris()

X = iris.data

y = iris.target

encoder = OneHotEncoder(sparse=False)

y\_onehot = encoder.fit\_transform(y.reshape(-1, 1))

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

model = tf.keras.Sequential([

tf.keras.layers.Input(shape=(4,)),

tf.keras.layers.Dense(10, activation='relu'),

tf.keras.layers.Dense(3, activation='softmax')

])

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

model.fit(X\_train, y\_train, epochs=100, batch\_size=8, validation\_split=0.1)

loss, accuracy = model.evaluate(X\_test, y\_test)

print("Test loss:", loss)

print("Test accuracy:", accuracy)

**OUT PUT :**

Epoch 1/100

11/11 [==============================] - 0s 10ms/step - loss: 1.1594 - accuracy: 0.3722 - val\_loss: 1.1288 - val\_accuracy: 0.3333

Epoch 2/100

11/11 [==============================] - 0s 4ms/step - loss: 1.0726 - accuracy: 0.3722 - val\_loss: 1.0634 - val\_accuracy: 0.3333

...

Epoch 100/100

11/11 [==============================] - 0s 4ms/step - loss: 0.3121 - accuracy: 0.9667 - val\_loss: 0.2591 - val\_accuracy: 1.0000

1/1 [==============================] - 0s 2ms/step - loss: 0.2207 - accuracy: 1.0000

Test loss: 0.22069558453559875

Test accuracy: 1.0