Ex No. 9	Implementation of Backpropagation Algorithm
Date:	

Aim

To build an artificial neural network by implementing backpropagation algorithm.

Definitions

Neural Network

Artificial neural networks, usually simply called neural networks or neural nets, are computing systems inspired by the biological neural networks that constitute animal brains. An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain.

Backpropagation Algorithm

Backpropagation is the essence of neural network training. It is the method of fine-tuning the weights of a neural network based on the error rate obtained in the previous epoch (i.e., iteration). Proper tuning of the weights allows you to reduce error rates and make the model reliable by increasing its generalization. Backpropagation in neural network is a short form for “backward propagation of errors.” It is a standard method of training artificial neural networks. This method helps calculate the gradient of a loss function with respect to all the weights in the network.

Procedure

Open PyCharm Community Edition.

Go to File menu → New Project → Specify the project name → Press “Create” button.

Right Click on Project name → New → Python File → Specify the file name → Press Enter.

Type the following codes. Right click on file name or coding window → Select “Run” to view the result.

Backprop.py

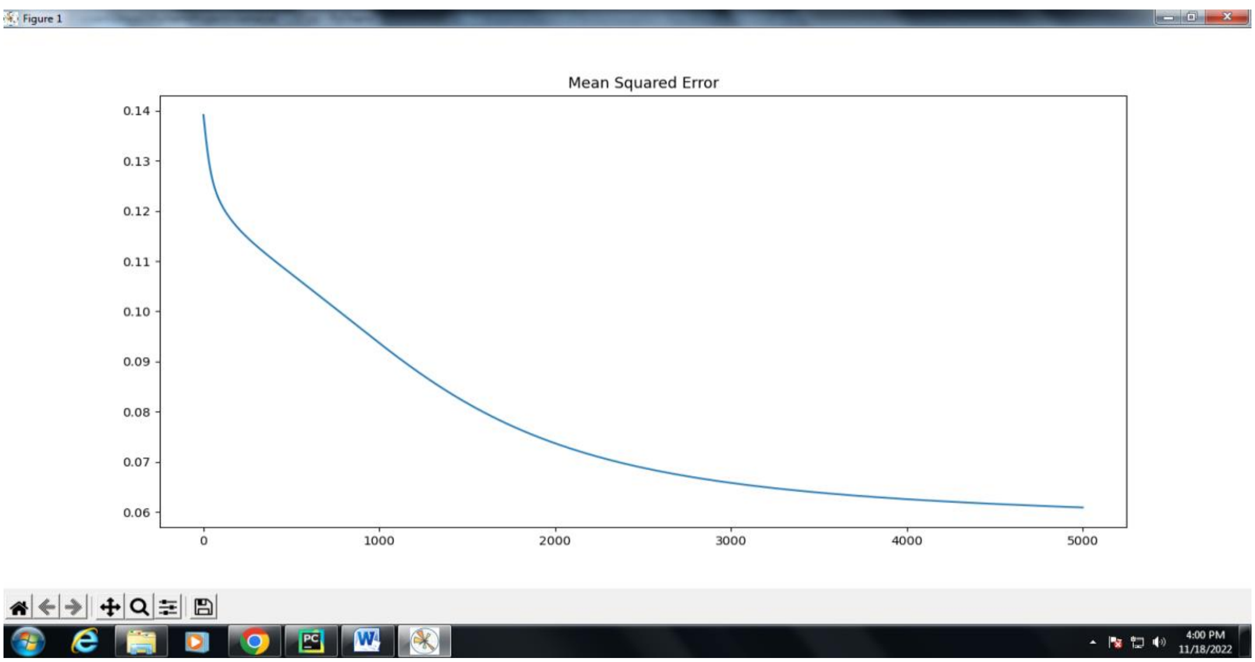
```
# Import Libraries
import numpy as np
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt

# Load dataset
data = load_iris()

# Get features and target
X=data.data
y=data.target

# Get dummy variable
y = pd.get_dummies(y).values

print(y[:3])

#Split data into train and test data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=20, random_state=4)

# Initialize variables
learning_rate = 0.1
iterations = 5000
N = y_train.size

# number of input features
input_size = 4

# number of hidden layers neurons
hidden_size = 2

# number of neurons at the output layer
output_size = 3

results = pd.DataFrame(columns=["mse", "accuracy"])

# Initialize weights
```

```
np.random.seed(10)

# initializing weight for the hidden layer
W1 = np.random.normal(scale=0.5, size=(input_size, hidden_size))

# initializing weight for the output layer
W2 = np.random.normal(scale=0.5, size=(hidden_size , output_size))

def sigmoid(x):
    return 1 / (1 + np.exp(-x))

def mean_squared_error(y_pred, y_true):
    return ((y_pred - y_true) ** 2).sum() / (2 * y_pred.size)

def accuracy(y_pred, y_true):
    acc = y_pred.argmax(axis=1) == y_true.argmax(axis=1)
    return acc.mean()

for itr in range(iterations):
    # feedforward propagation
    # on hidden layer
    Z1 = np.dot(X_train, W1)
    A1 = sigmoid(Z1)

    # on output layer
    Z2 = np.dot(A1, W2)
    A2 = sigmoid(Z2)

    # Calculating error
    mse = mean_squared_error(A2, y_train)
    acc = accuracy(A2, y_train)
    results = results.append({"mse": mse, "accuracy": acc}, ignore_index=True)

    # backpropagation
    E1 = A2 - y_train
    dW1 = E1 * A2 * (1 - A2)

    E2 = np.dot(dW1, W2.T)
    dW2 = E2 * A1 * (1 - A1)

    # weight updates
    W2_update = np.dot(A1.T, dW1) / N
    W1_update = np.dot(X_train.T, dW2) / N

    W2 = W2 - learning_rate * W2_update
    W1 = W1 - learning_rate * W1_update

print(results.mse.plot(title="Mean Squared Error"))
plt.show()
print(results.accuracy.plot(title="Accuracy"))
plt.show()
```

```
# feedforward
Z1 = np.dot(X_test, W1)
A1 = sigmoid(Z1)

Z2 = np.dot(A1, W2)
A2 = sigmoid(Z2)

acc = accuracy(A2, y_test)
print("Accuracy: {}".format(acc))
```

Output

```
C:\Users\2mca1\PycharmProjects\sumaiya\venv\Scripts\python.exe
C:/Users/2mca1/PycharmProjects/sumaiya/backprop.py

[[1 0 0]

 [1 0 0]

 [1 0 0]]

AxesSubplot(0.125,0.11;0.775x0.77)

AxesSubplot(0.125,0.11;0.775x0.77)

Accuracy: 0.8

Process finished with exit code 0
```



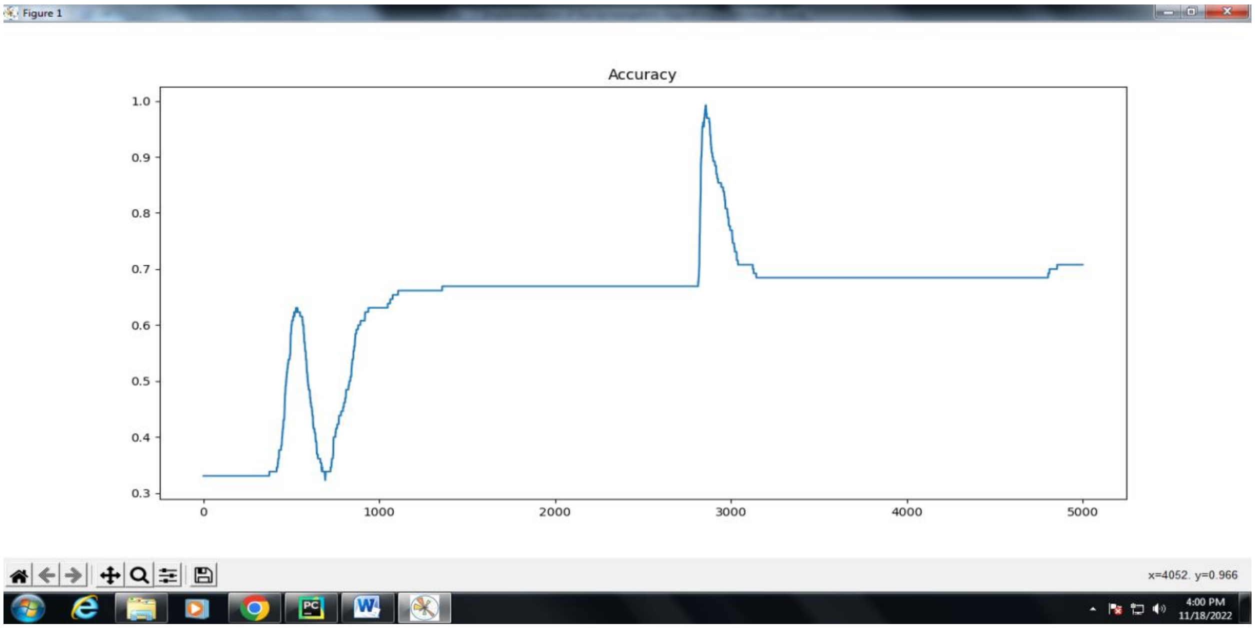


Figure 1
Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
1000
2000
3000
4000
5000
x=4052. y=0.966
4:00 PM
11/18/2022

Result

Thus, an artificial neural network has been constructed by implementing backpropagation algorithm.