

Name – Umar Momin

PRN -20210812016

DNN

## Backward Propagation using tensorflow

**AIM-** write a program in python for Backward propagation using tensorflow

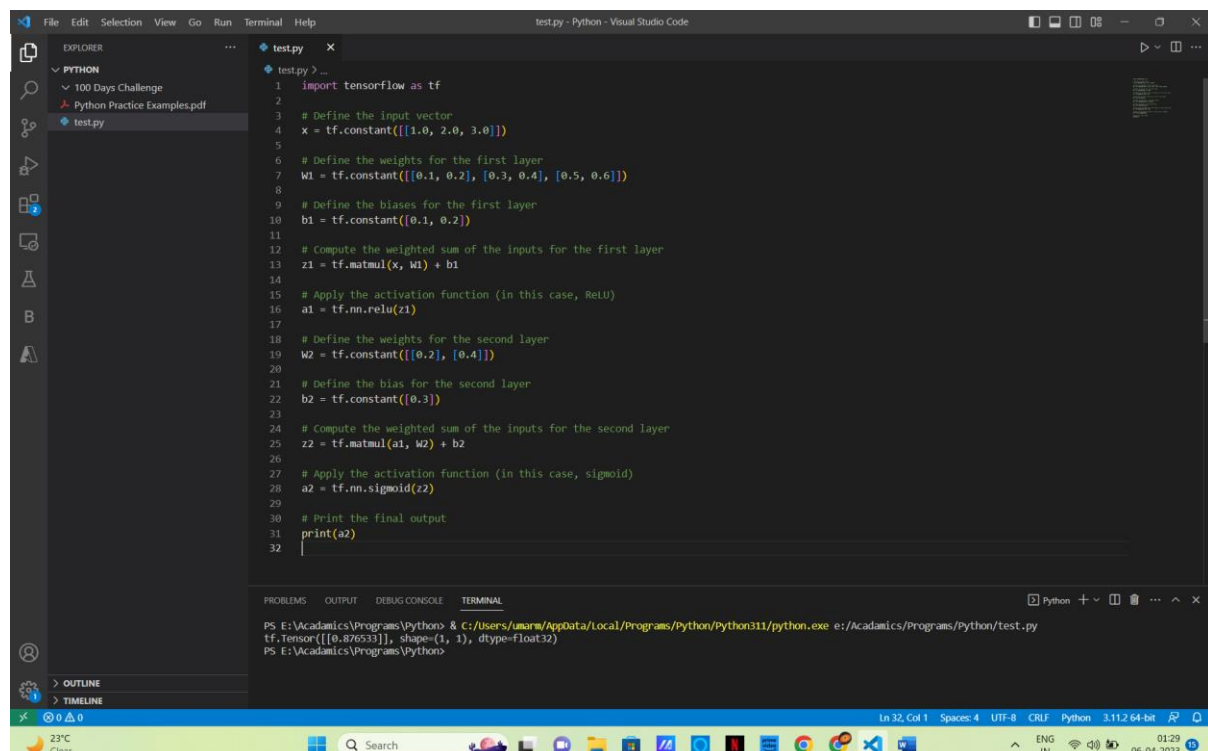
### Theory-

Backward propagation is a key concept in artificial neural networks and is used to calculate the gradients of the error with respect to the weights and biases of the network. The process involves working backwards from the output layer to the input layer of the network, using the chain rule of calculus to calculate the derivative of the error with respect to each weight and bias.

To implement backward propagation, we first calculate the error between the actual outputs and the desired outputs. Then, we calculate the partial derivative of the error with respect to each weight and bias at each layer of the network. We use these partial derivatives to update the weights and biases in the network, using a gradient descent algorithm to move towards the optimal values.

In summary, backward propagation is a fundamental component of machine learning algorithms, allowing us to train neural networks to accurately predict output values from input data.

### Program-



```
test.py - Python - Visual Studio Code
1 import tensorflow as tf
2
3 # Define the input vector
4 x = tf.constant([[1.0, 2.0, 3.0]])
5
6 # Define the weights for the first layer
7 w1 = tf.constant([[0.1, 0.2], [0.3, 0.4], [0.5, 0.6]])
8
9 # Define the biases for the first layer
10 b1 = tf.constant([0.1, 0.2])
11
12 # Compute the weighted sum of the inputs for the first layer
13 z1 = tf.matmul(x, w1) + b1
14
15 # Apply the activation function (in this case, ReLU)
16 a1 = tf.nn.relu(z1)
17
18 # Define the weights for the second layer
19 w2 = tf.constant([[0.2], [0.4]])
20
21 # Define the bias for the second layer
22 b2 = tf.constant([0.3])
23
24 # Compute the weighted sum of the inputs for the second layer
25 z2 = tf.matmul(a1, w2) + b2
26
27 # Apply the activation function (in this case, sigmoid)
28 a2 = tf.nn.sigmoid(z2)
29
30 # Print the final output
31 print(a2)
32
```

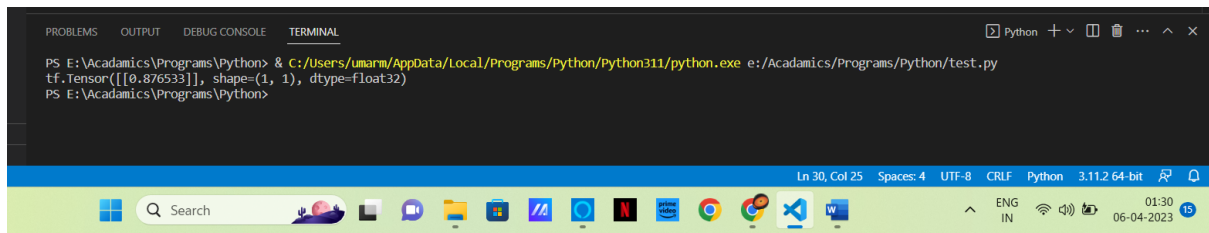
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS E:\Academics\Programs\Python> & C:\Users\umam\AppData\Local\Programs\Python\Python311\python.exe e:/Academics/Programs/Python/test.py  
tf.Tensor([[0.876533]], shape=(1, 1), dtype=float32)  
PS E:\Academics\Programs\Python>

23°C Clear

Ln 32, Col 1, Spaces: 4, UTF-8, CRLF, Python, 3.11.2 64-bit

01:29  
06-04-2023



The image shows a Windows terminal window with a dark background. The title bar at the top indicates the file is 'Python'. The terminal content shows a command prompt session where a Python script is executed. The output displays a TensorFlow tensor. The Windows taskbar is visible at the bottom, showing various application icons and the system clock.

```
PS E:\Academics\Programs\Python> & C:/Users/umarm/AppData/Local/Programs/Python/Python311/python.exe e:/Academics/Programs/Python/test.py
tf.Tensor([[0.876533]], shape=(1, 1), dtype=float32)
PS E:\Academics\Programs\Python>
```

Conclusion – Here in this practical we have performed program for Backword propagation using tensorflow