

Name – Umar Momin

PRN – 20210812016

Subject – DNN

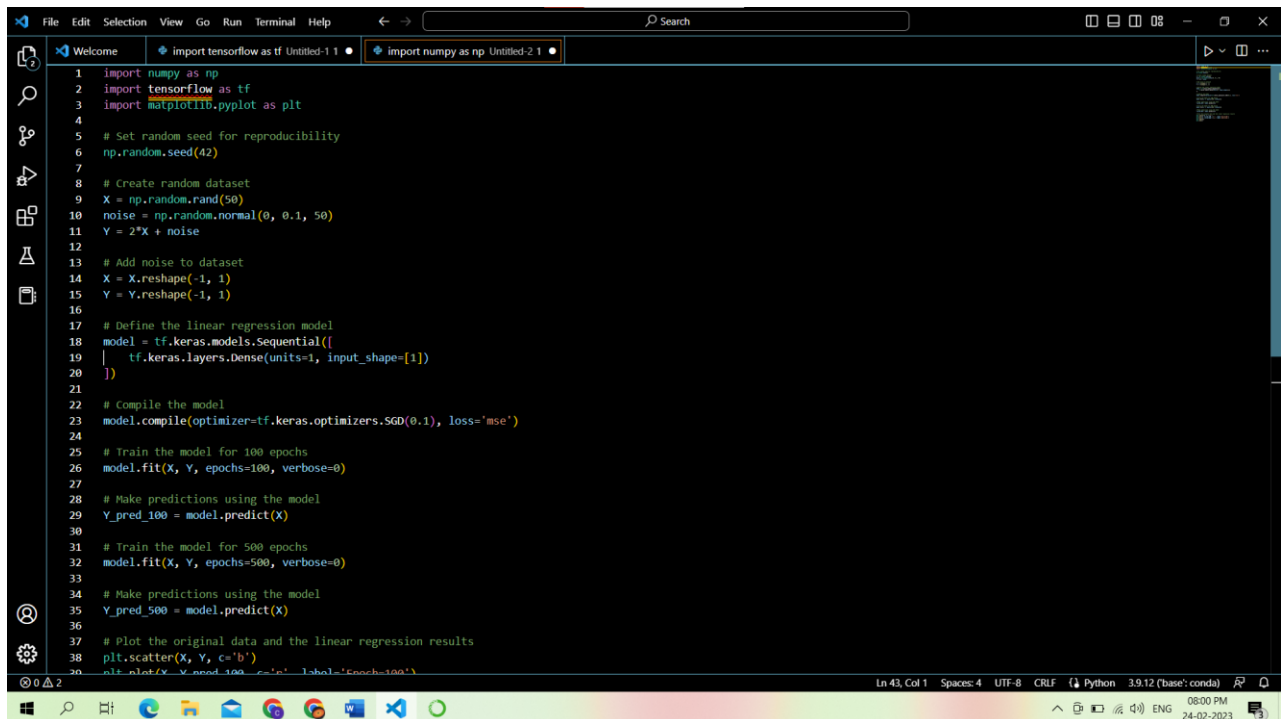
ASSIGNMENT

Creating a Linear Regression Model as an ANN with TensorFlow

Assignment 1

- Write a program to create a random 50 data points (X,Y) plot data points and add some noise in data set and apply linear regression use epoch=100 and check the result ,epoch=500 check result and plot graph.

Code



```
1 import numpy as np
2 import tensorflow as tf
3 import matplotlib.pyplot as plt
4
5 # Set random seed for reproducibility
6 np.random.seed(42)
7
8 # Create random dataset
9 X = np.random.rand(50)
10 noise = np.random.normal(0, 0.1, 50)
11 Y = 2*X + noise
12
13 # Add noise to dataset
14 X = X.reshape(-1, 1)
15 Y = Y.reshape(-1, 1)
16
17 # Define the linear regression model
18 model = tf.keras.models.Sequential([
19     tf.keras.layers.Dense(units=1, input_shape=[1])
20 ])
21
22 # Compile the model
23 model.compile(optimizer=tf.keras.optimizers.SGD(0.1), loss='mse')
24
25 # Train the model for 100 epochs
26 model.fit(X, Y, epochs=100, verbose=0)
27
28 # Make predictions using the model
29 Y_pred_100 = model.predict(X)
30
31 # Train the model for 500 epochs
32 model.fit(X, Y, epochs=500, verbose=0)
33
34 # Make predictions using the model
35 Y_pred_500 = model.predict(X)
36
37 # Plot the original data and the linear regression results
38 plt.scatter(X, Y, c='b')
39 plt.plot(X, Y_pred_100, c='r', label='Epoch=100')
40 plt.plot(X, Y_pred_500, c='g', label='Epoch=500')
```

```
File Edit Selection View Go Run Terminal Help
import tensorflow as tf
import numpy as np

# Compile the model
model.compile(optimizer=tf.keras.optimizers.SGD(0.1), loss='mse')

# Train the model for 100 epochs
model.fit(X, Y, epochs=100, verbose=0)

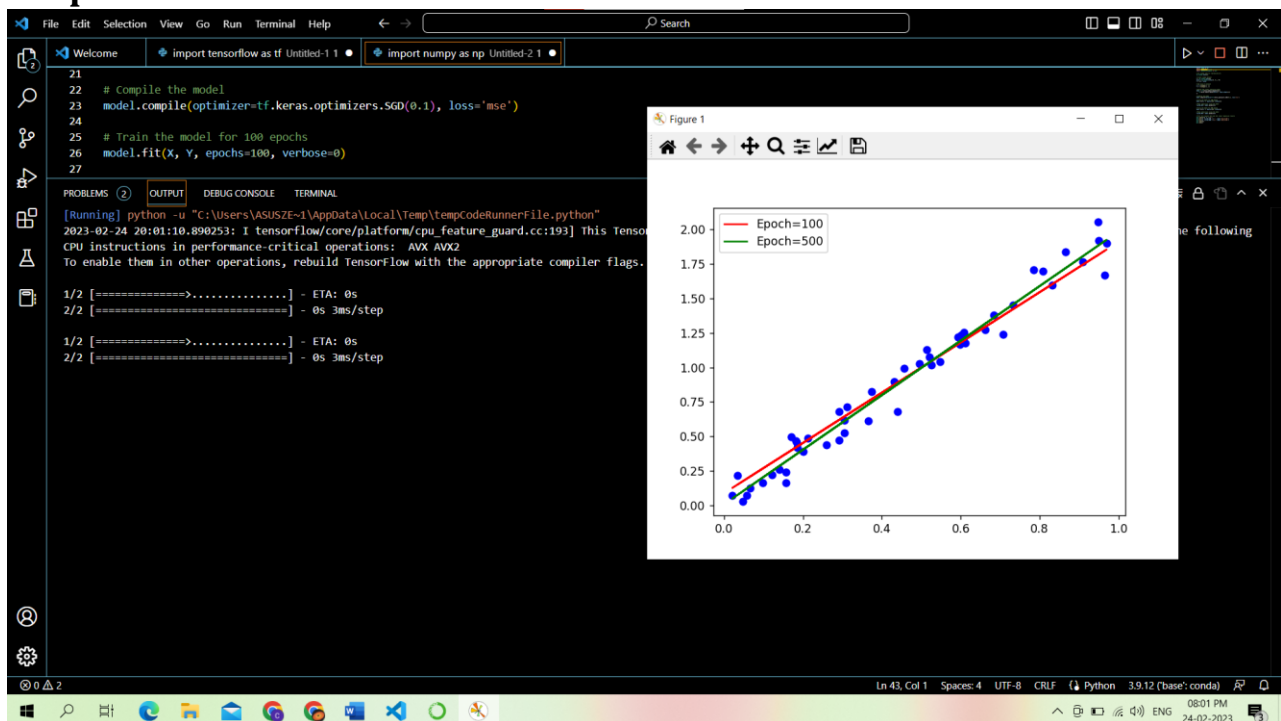
# Make predictions using the model
Y_pred_100 = model.predict(X)

# Train the model for 500 epochs
model.fit(X, Y, epochs=500, verbose=0)

# Make predictions using the model
Y_pred_500 = model.predict(X)

# Plot the original data and the linear regression results
plt.scatter(X, Y, c='b')
plt.plot(X, Y_pred_100, c='r', label='Epoch=100')
plt.plot(X, Y_pred_500, c='g', label='Epoch=500')
plt.legend()
plt.show()
```

Output –



Assignment 2 –

Write a program to creating a Linear Regression Model as an ANN with TensorFlow by using following data set.”

Code –

The screenshot shows a Jupyter Notebook environment with the following code:

```

1 import numpy as np
2 import tensorflow as tf
3 import matplotlib.pyplot as plt
4
5 # Define the dataset
6 X = np.array([1, 2, 3, 4, 5, 6])
7 Y = np.array([2, 4, 6, 8, 10, 12])
8
9 # Define the linear regression model as an ANN
10 model = tf.keras.models.Sequential([
11     tf.keras.layers.Dense(units=1, input_shape=[1])
12 ])
13
14 # Compile the model
15 model.compile(optimizer=tf.keras.optimizers.SGD(0.1), loss='mse')
16
17 # Train the model
18 history = model.fit(X, Y, epochs=50, verbose=0)
19
20 # Make predictions using the model
21 Y_pred = model.predict(X)
22
23 # Print the model's summary
24 model.summary()
25
26 # Plot the original data and the linear regression result
27 plt.scatter(X, Y, c='b')
28 plt.plot(X, Y_pred, c='r')
29 plt.show()
30

```

The interface includes a top menu bar (File, Edit, Selection, View, Go, Run, Terminal, Help), a toolbar with icons for file operations and execution, and a bottom status bar showing the current cell (Ln 30, Col 1), encoding (UTF-8), and environment (Python 3.9.12 (base: conda)).

Output –

