EX.NO:8 **DATE:4/9/2024**

Reg.no:220701025

IMPLEMENTING ARTIFICIAL NEURAL NETWORKS FOR AN APPLICATION USING PYTHON – REGRESSION

AIM :

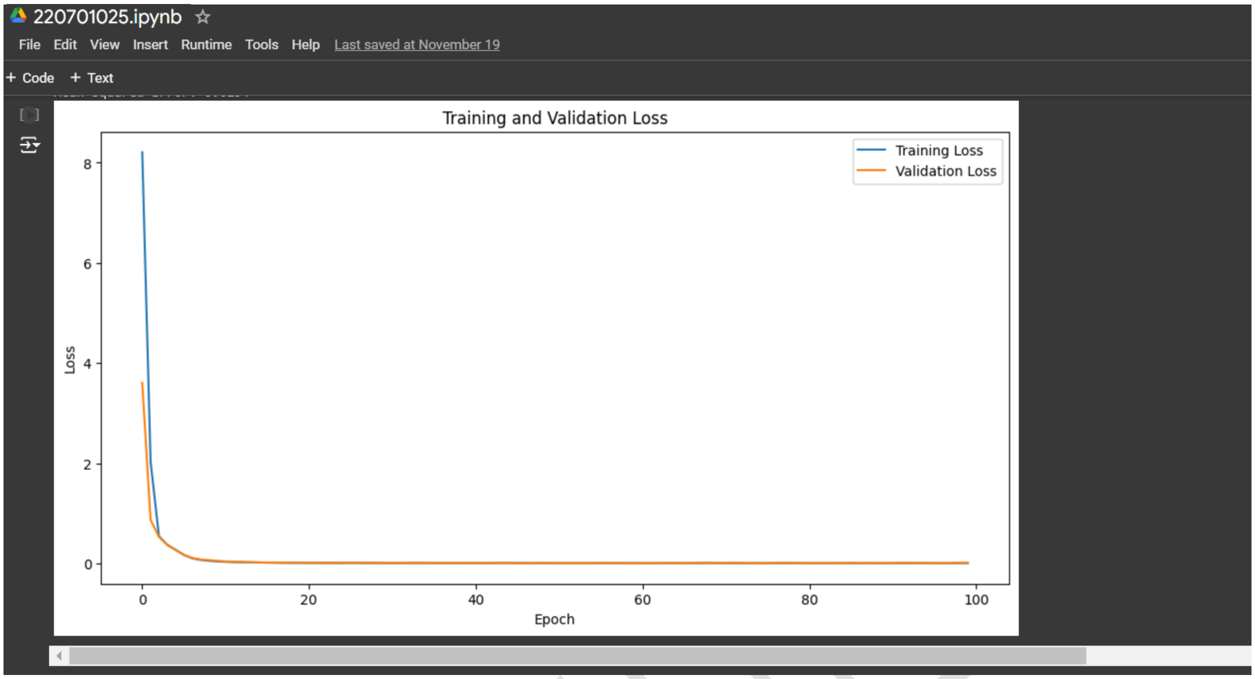
To implementing artificial neural networks for an application in Regression using python.

CODE:

```
import numpy as np

import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from keras.models import Sequential

from keras.layers import Dense

from keras.optimizers import Adam

import matplotlib.pyplot as plt

np.random.seed(42)

X = np.random.rand(1000, 3)  # 1000 samples, 3 features

y = 3 * X[:, 0] + 2 * X[:, 1] ** 2 + 1.5 * np.sin(X[:, 2] * np.pi) +
np.random.normal(0, 0.1, 1000)  # Non-linear relationship

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

scaler = StandardScaler()

X_train = scaler.fit_transform(X_train)

X_test = scaler.transform(X_test)

model = Sequential()
```

```
model.add(Dense(10, input_dim=X_train.shape[1], activation='relu'))

model.add(Dense(10, activation='relu'))

model.add(Dense(1, activation='linear'))

model.compile(optimizer=Adam(learning_rate=0.01),
loss='mean_squared_error')

history = model.fit(X_train, y_train, epochs=100, batch_size=32,
validation_split=0.2, verbose=1)

y_pred = model.predict(X_test)

mse = np.mean((y_test - y_pred.flatten()) ** 2)

print(f'Mean Squared Error: {mse:.4f}')

plt.figure(figsize=(12, 6))

plt.plot(history.history['loss'], label='Training Loss')

plt.plot(history.history['val_loss'], label='Validation Loss')

plt.title('Training and Validation Loss')

plt.xlabel('Epoch')

plt.ylabel('Loss')

plt.legend()

plt.show()
```

OUTPUT:


220701025.ipynb ☆
File Edit View Insert Runtime Tools Help Last saved at November 19
+ Code + Text
Training and Validation Loss
Training Loss
Validation Loss
Loss
0
2
4
6
8
0
20
40
60
80
100
Epoch