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

Reg.no:220701017

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 matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from sklearn.preprocessing import StandardScaler

np.random.seed(42)

X = np.linspace(0, 10, 100)
y = 2 * X + 1 + np.random.normal(0, 1, 100)

X = X.reshape(-1, 1)


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(units=64, activation='relu', input_dim=1))
```

```
model.add(Dense(units=32, activation='relu'))

model.add(Dense(units=1))

model.compile(optimizer='adam', loss='mean_squared_error')

history = model.fit(X_train, y_train, epochs=100, batch_size=10,
validation_split=0.2)

y_pred = model.predict(X_test)

plt.scatter(X_test, y_test, color='blue', label='True values')
plt.scatter(X_test, y_pred, color='red', label='Predictions')
plt.plot(X_test, y_pred, color='red', linewidth=2)
plt.title('Artificial Neural Network Regression')
plt.xlabel('X')
plt.ylabel('y')
plt.legend()
plt.show()

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('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
```

OUTPUT: