DL – 6 BOSTON HOUSING PRICE

```
pip install scikit-learn==1.1.3

import numpy as np
from sklearn.datasets import load_boston
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import tensorflow as tf
from tensorflow import keras
from sklearn.preprocessing import StandardScaler

boston = load_boston()
X, y = boston.data, boston.target

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

linear_model = LinearRegression()

linear_model.fit(X_train, y_train)

linear_predictions = linear_model.predict(X_test)
linear_mse = mean_squared_error(y_test, linear_predictions)
print('Linear Regression Mean Squared Error:', linear_mse)

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

model = keras.Sequential([
   keras.layers.Dense(64, activation='relu', input_shape=(X_train.shape[1],)),
   keras.layers.Dense(64, activation='relu'),
   keras.layers.Dense(1)
])

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

model.fit(X_train_scaled, y_train, epochs=100, batch_size=32, verbose=1)

mse = model.evaluate(X_test_scaled, y_test)
print('Deep Neural Network Mean Squared Error:', mse)

predictions = model.predict(X_test_scaled)

for i in range(10):
   print('Predicted Price:', predictions[i][0], 'Actual Price:', y_test[i])
```

EXPLANATION OF CODE

Let me explain the code step by step:

1. The code starts by importing the necessary libraries: `pandas` for data manipulation, `numpy` for numerical computations, `tensorflow` for creating and training the model, and `sklearn` for loading the Boston dataset and performing data preprocessing.

2. The Boston dataset is loaded using the `load_boston()` function from `sklearn.datasets`. The dataset contains information about housing prices in Boston.

3. The loaded dataset is stored in a pandas DataFrame called `data`, and the target variable (housing prices) is stored in the `targets` variable.

4. Data preprocessing is performed using `StandardScaler()` from `sklearn.preprocessing`. It standardizes the features of the dataset by subtracting the mean and scaling to unit variance. The standardized data is stored in the `data_scaled` variable.

5. The number of features in the dataset is determined by `data.shape[1]`, and it is stored in the `n_features` variable.

6. The model is defined using `tf.keras.Sequential()`, which allows creating a sequential model in TensorFlow. The model consists of three layers: two hidden layers with 64 neurons each and ReLU activation function, and an output layer with a single neuron (for predicting housing prices).

7. The model is compiled using `model.compile()`. Here, the Adam optimizer with a learning rate of 0.1 is used, and the mean squared error (MSE) is chosen as the loss function. Additionally, mean absolute error (MAE) is specified as a metric to evaluate the model.

8. The model is trained using `model.fit()`. The `train_data` and `train_target` are used for training the model, and the training is performed for 10 epochs. The `validation_split=0.2` parameter means that 20% of the training data will be used for validation during training. The training history is stored in the `history` variable.

9. After training, the model is evaluated using `model.evaluate()`. The `test_data` and `test_target` are used to compute the test loss and test MAE. The test MAE is then printed.

In summary, the code loads the Boston housing dataset, pre-processes the data by standardizing it, defines a neural network model with two hidden layers, compiles the model with appropriate optimizer and loss function, trains the model on the training data, and evaluates the model on the test data. The test MAE is then printed as a measure of the model's performance.

Questions

1. What is the assignment about – Linear regression by using Deep Neural network: Implement Boston housing price. prediction problem by Linear regression using Deep Neural network. Use Boston House price prediction data set.

2. What is linear regression – Linear regression is a statistical approach that is commonly used to model the relationship between a dependent variable and one or more independent variables. It assumes a linear relationship between the variables and uses mathematical methods to estimate the coefficients that best fit the data.

3. Deep neural networks - They are a type of machine learning algorithm that are modeled after the structure and function of the human brain. They consist of multiple layers of

interconnected neurons that process data and learn from it to make predictions. Boston House Price Prediction is a common example used to illustrate how a deep neural network can work for regression tasks. The goal of this task is to predict the price of a house in Boston based on various features such as the number of rooms, crime rate, and accessibility to public transportation.

4. Dataset used? - Boston House Price Prediction Dataset- Boston House Price Prediction is a well-known dataset in machine learning and is often used to demonstrate regression analysis techniques. The dataset contains information about 506 houses in Boston, Massachusetts, USA. The goal is to predict the median value of owner-occupied homes in thousands of dollars. The dataset includes 13 input features, which are: CRIM: per capita crime rate by town , ZN: proportion of residential land zoned for lots over 25,000 sq.ft.

5. After the model is trained, it can be used to predict the median value of owner-occupied homes based on the input features. The model's accuracy can be evaluated using metrics such as mean squared error or mean absolute error.