EX-01 Developing a Neural Network Regression Model

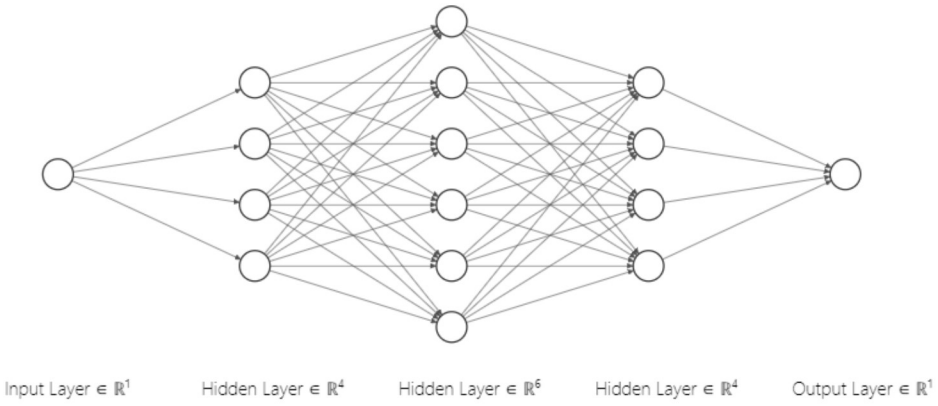
Aim:

To develop a neural network regression model for the given dataset.

Theory:

- Design and implement a neural network regression model to accurately predict a continuous target variable based on a set of input features within the provided dataset.
- The objective is to develop a robust and reliable predictive model that can capture complex relationships in the data, ultimately yielding accurate and precise predictions of the target variable.
- The model should be trained, validated, and tested to ensure its generalization capabilities on unseen data, with an emphasis on optimizing performance metrics such as mean squared error or mean absolute error.

Neural Network Model:



Design Steps:

- STEP 1:Loading the dataset
- STEP 2:Split the dataset into training and testing
- STEP 3:Create MinMaxScalar objects ,fit the model and transform the data.
- STEP 4:Build the Neural Network Model and compile the model.
- STEP 5:Train the model with the training data.
- STEP 6:Plot the performance plot
- STEP 7:Evaluate the model with the testing data.

Name : SHARAN MJ

RegNo : 212222240097

Program:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

df=pd.read_excel('/content/owndata.xlsx')
df = df.astype({'input':'float'})
df = df.astype({'output':'float'})
df

x=df[['input']].values
y=df[['output']].values
x

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33,random_state=33)
scalar=MinMaxScaler()
scalar.fit(x_train)

x_train1=scalar.transform(x_train)
ai=Sequential([
    Dense (units = 8, activation = 'relu'),
    Dense (units = 10, activation = 'relu'),
    Dense (units = 1)])

ai.compile(optimizer='rmsprop',loss='mse')
ai.fit(x_train1,y_train,epochs=2000)

loss_df = pd.DataFrame(ai.history.history)
loss_df.plot()

X_test1 = scalar.transform(x_test)
ai.evaluate(X_test1,y_test)

X_n1 = [[float(input('enter the value : '))]]
X_n1_1 = scalar.transform(X_n1)
a=ai.predict(X_n1_1)
print('The predicted output : ',a)
```

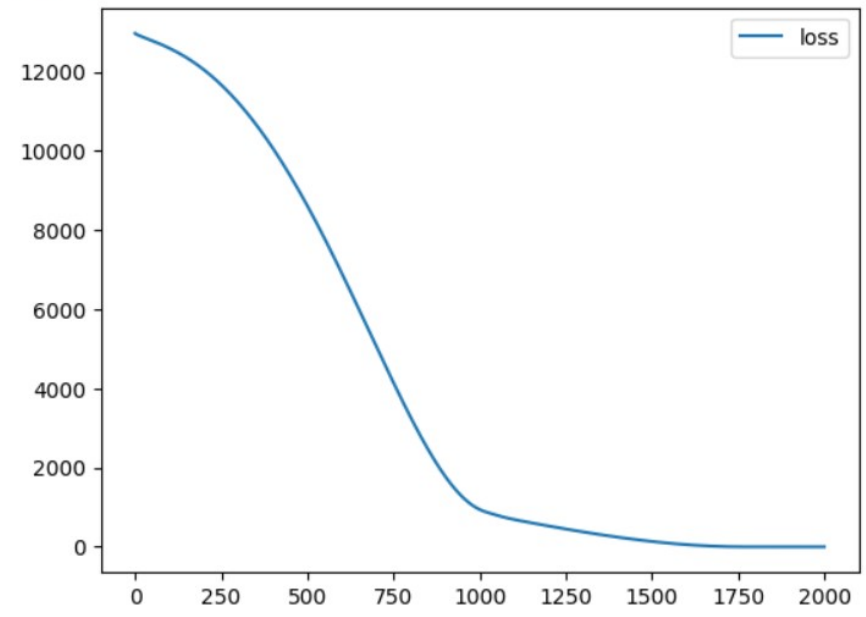
Output:

Dataset:

	input	output
0	1.0	11.0
1	2.0	21.0
2	3.0	31.0
3	4.0	41.0
4	5.0	51.0
5	6.0	61.0
6	7.0	71.0
7	8.0	81.0
8	9.0	91.0
9	10.0	101.0
10	11.0	111.0
11	12.0	121.0
12	13.0	131.0
13	14.0	141.0
14	15.0	151.0
15	16.0	161.0
16	17.0	171.0
17	18.0	181.0
18	19.0	191.0
19	20.0	201.0

Training Loss Vs Iteration Plot:

```
<Axes: >
```



Epoch:

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step - loss: 0.0191
Epoch 1985/2000
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 58ms/step - loss: 0.0192
Epoch 1986/2000
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step - loss: 0.0191
Epoch 1987/2000
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 71ms/step - loss: 0.0192
Epoch 1988/2000
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - loss: 0.0191
Epoch 1989/2000
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 132ms/step - loss: 0.0191
Epoch 1990/2000
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - loss: 0.0191
Epoch 1991/2000
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 138ms/step - loss: 0.0191
Epoch 1992/2000
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 84ms/step - loss: 0.0191
Epoch 1993/2000
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 61ms/step - loss: 0.0191
Epoch 1994/2000
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - loss: 0.0191
Epoch 1995/2000
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step - loss: 0.0191
Epoch 1996/2000
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 67ms/step - loss: 0.0191
Epoch 1997/2000
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - loss: 0.0191
Epoch 1998/2000
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 72ms/step - loss: 0.0191
Epoch 1999/2000
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 58ms/step - loss: 0.0191
Epoch 2000/2000
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - loss: 0.0191
<keras.src.callbacks.history.History at 0x7b7a2f299780>
```

Test Data Root Mean Squared Error:

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 115ms/step - loss: 0.7489
0.7489051818847656
```

New Sample Data Prediction:

```
enter the value : 7
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
The predicted output :  [[70.91237]]
```

Result:

Thus a neural network regression model for the given dataset is developed and the prediction for the given input is obtained accurately.