

# Developing a Neural Network Regression Model

## AIM

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

## THEORY

Explain the problem statement

## Neural Network Model

Include the neural network model diagram.

## 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.

## 🔗 PROGRAM

```
import pandas as pd
from google.colab import auth
import gspread
from google.auth import default

auth.authenticate_user()
creds, _ = default()
gc = gspread.authorize(creds)

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
import tensorflow as tf
tf.__version__

worksheet = gc.open('firstdataset').sheet1
rows = worksheet.get_all_values()

df = pd.DataFrame(rows[1:], columns=rows[0])
df.head(n=10)
df.dtypes
df = df.astype({'X': 'float'})
df = df.astype({'Y': 'float'})
df.dtypes
X = df[['X']].values
X
Y = df[['Y']].values
Y
X_train,X_test,Y_train,Y_test =
train_test_split(X,Y,test_size=0.33,random_state=50)
X_test.shape
X_train
scaler = MinMaxScaler()
scaler.fit(X_train)
X_train_scaled = scaler.transform(X_train)
X_train_scaled

ai_brain = Sequential([
```

```

        Dense(2,activation = 'relu'),
        Dense(1,activation = 'relu')
    ])
    ai_brain.compile(optimizer = 'rmsprop',loss = 'mse')
    ai_brain.fit(x = X_train_scaled,y = Y_train,epochs = 20000)
    loss_df = pd.DataFrame(ai_brain.history.history)
    loss_df.plot()
    X_test
    X_test_scaled = scaler.transform(X_test)
    X_test_scaled
    ai_brain.evaluate(X_test_scaled,Y_test)

input = [[120]]
input_scaled = scaler.transform(input)
input_scaled.shape
input_scaled
ai_brain.predict(input_scaled)

```

## 🔗 Dataset Information

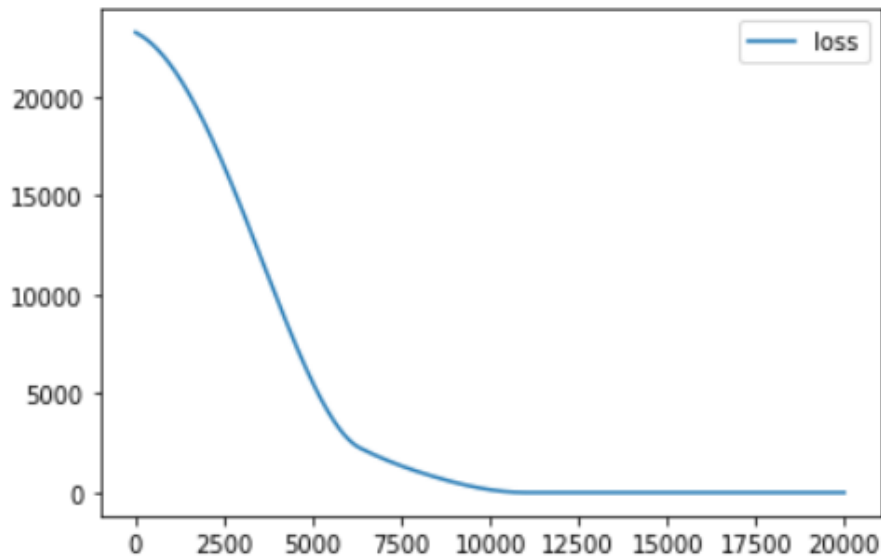
	A	B	C
1	X	Y	
2	10	35	
3	20	65	
4	30	95	
5	40	125	
6	50	155	
7	60	185	
8	70	215	
9	80	245	
10	90	275	
11			
12			

## 🔗 OUTPUT

## 🔗 Training Loss Vs Iteration Plot

```
loss_df.plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa74e055fd0>
```



## Test Data Root Mean Squared Error

```
ai_brain.evaluate(X_test_scaled,Y_test)
```

```
1/1 [=====] - 0s 126ms/step - loss: 7.8103e-04  
0.0007810332463122904
```

## New Sample Data Prediction

### sample input

```
input = [[120]]
```

### sample output

```
ai_brain.predict(input_scaled)
```

```
array([[364.95758]], dtype=float32)
```

## RESULT

Successfully a neural network regression model is performed for the given dataset.