# EXP:01Developing a Neural Network Regression Model

## AIM

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

## THEORY The problem statement revolves around creating a predictive model using a neural network to establish a relationship between input and output data. The dataset consists of input-output pairs stored in a Google Sheets document. The goal is to design a neural network that learns from the input-output patterns in the dataset and can predict the output for new, unseen input values. The input data represents the 'input' variable, and the corresponding output data represents the 'output' variable. The dataset is divided into training and testing sets for model evaluation. The neural network architecture includes multiple layers, such as the input layer, two hidden layers with rectified linear unit (ReLU) activation functions, and an output layer. The model is trained using the mean squared error (MSE) loss function and the RMSprop optimizer. The training progress, as well as the final model's performance on the testing set, is visualized and evaluated. Ultimately, the trained neural network is intended to predict the output for new input values, which is demonstrated by predicting the output for an example input of 4 after appropriate data preprocessing.

## 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.

## PROGRAM

from google.colab import auth

import gspread

from google.auth import default

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

auth.authenticate\_user()

creds, \_ = default()

gc = gspread.authorize(creds)

worksheet = gc.open('data').sheet1

rows = worksheet.get\_all\_values()

df = pd.DataFrame(rows[1:], columns=rows[0])

df1 = df.astype({'input':'float'})

df1 = df.astype({'output':'float'})

df1.head()

x = df1[['input']].values

y = df1[['output']].values

x

y

x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y,test\_size=0.33,random\_state=33)

scaler = MinMaxScaler()

scaler.fit(x\_train)

x\_train1 = scaler.transform(x\_train)

x\_train1 = scaler.transform(x\_train)

ai\_brain = Sequential([

Dense(8,activation='relu'),

Dense(10,activation='relu'),

Dense(1)

])

ai\_brain.compile(optimizer='rmsprop',loss = 'mse')

ai\_brain.fit(x\_train1,y\_train,epochs=2000)

loss\_df = pd.DataFrame(ai\_brain.history.history)

loss\_df.plot()

x\_test1 = scaler.transform(x\_test)

ai\_brain.evaluate(x\_test1,y\_test)

x\_n1 = [[4]]

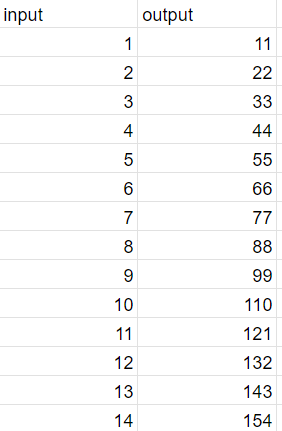
x\_n1\_1 = scaler.transform(x\_n1)

ai\_brain.predict(x\_n1\_1)

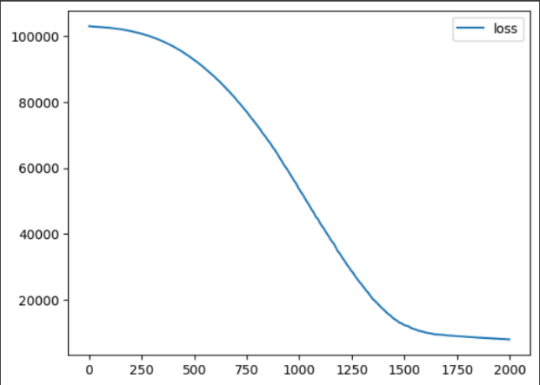
## 

## OUTPUT:

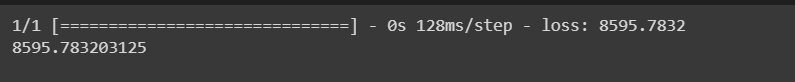
### SAMPLE DATA:

[](https://github.com/Prethiveerajan/basic-nn-model/blob/main/data.png)

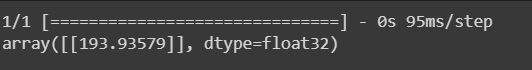
### Training Loss Vs Iteration Plot

[](https://github.com/Prethiveerajan/basic-nn-model/blob/main/graph.png)

### Test Data Root Mean Squared Error:

[](https://github.com/Prethiveerajan/basic-nn-model/blob/main/test.png)

### New Sample Data Prediction:

[](https://github.com/Prethiveerajan/basic-nn-model/blob/main/predict.png)

## RESULT

Thus a neural network regression model for the given dataset is written and executed successfully.