భారతీయ సాంకేతిక విజ్ఞాన సంస్థ హైదరాబాద్
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

An Assignment Report

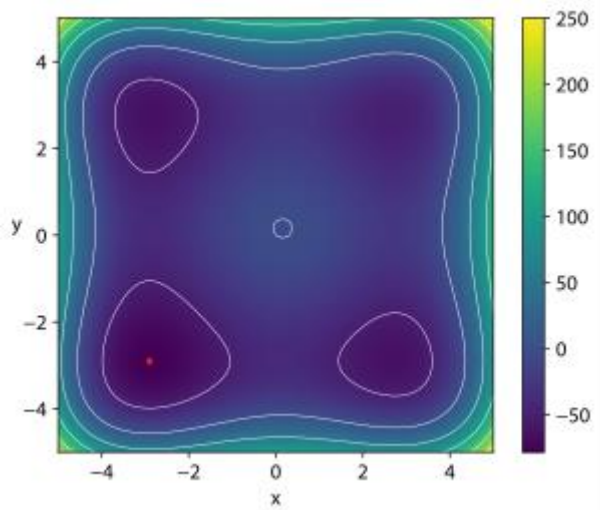
Training of Artificial Neural Network

Submitted by:

ASHISH KUMAR SINGH

PB22MTECH11003

The given test function is **Styblinski-Tang Function** for **n=2**



$$f(\boldsymbol{x}) = \frac{\sum_{i=1}^{n} x_i^4 - 16x_i^2 + 5x_i}{2}$$

$$-39.16617n < f(\underbrace{-2.903534, \ldots, -2.903534}_{n \text{ times}}) < -39.16616n$$

$$-5 \leq x_i \leq 5$$

Python Implementation of the given function for **N=2** is

z = 0.5 * ((x4 + x**4) - 16 * (x**2 + x**2)+ 5 * (x + x))** & **range (-5,5)**

Generating Dataset for given function: -

- Importing **numpy** library
- Creating an empty array for storing the value of **"X"** range between (-5,5) of step size of 0.004
- Creating an empty array for storing the function value **"Z"**
- Creating a dataframe for the data generated above as

  data=zip (X, Y, Z) **data_to_excel=pd.DataFrame(data)**

- Coverting **data=zip (X, Y, Z)** in to excel file and giving the path location.

Reading the Data from the saved excel file: -

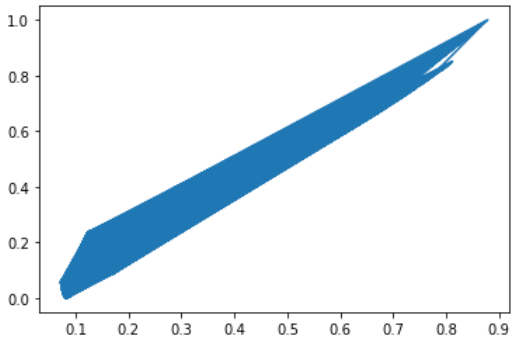
- Importing pandas library
- Providing the path of the source file to the **data=pd.read_excel("")**

Applying the Artificial Neural Network:-

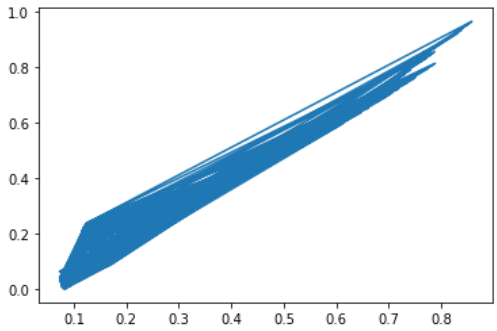
- Normalizing the data between **0 – 1** for that import **MinMaxScaler** function from **sklearn.preprocessing**
- Dividing the data in to input and output, the first and second column as the input and the third column function value as the output by **scaled_data[: , :]** function.
- Importing the **train_test_split** function from **sklearn.model_selection**
- Splitting the data in to train and test data set by **train_test_split(X,y,test_size=0.15,random_state=0)** function.
- Importing the **Sequential** function from **tensorflow.keras**
- Importing the **Dense** function from **tensorflow.keras.layers**
- Applying the ANN model by selecting the number of **hidden layers,** number of **nodes** and providing the **Activation function , optimizer function, loss function.**
- Providing **epochs** value and **validation_split** value.
- Import **r2_score** function from **sklearn.metrics**
- Finding out **Mean squared error for training data** **mse_train=np.square(y_train_pre - y_train).mean()**
- Finding out the r2 value of test and train data by **r2_score(y_test,y_test_pre)** and **r2_score(y_train,y_train_pre)** function.
- Importing **matplotlib.pyplot** library for plotting the graph.
- Plotting the grapgh for test and test predicted value by **ptl.plot(y_test_pre,y_test)** function.
- Plotting the grapgh for traint and train predicted value by **ptl.plot(y_train_pre,y_train)** function.

Effect of changing the optimizer

Optimizer = "adm"

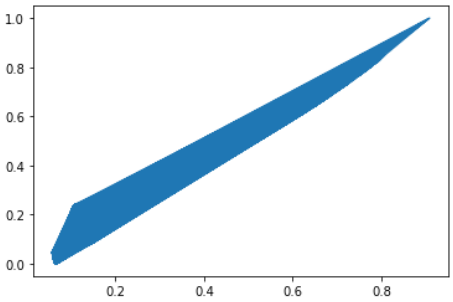


R2 for train & train predicted =**0.89**

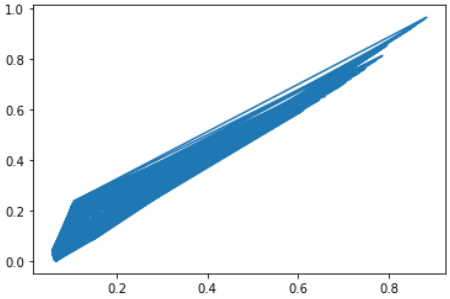


R2 for test & test predicted =**0.91**

Optimizer = "RMSProp"



R2 for train & train predicted = **0.88**

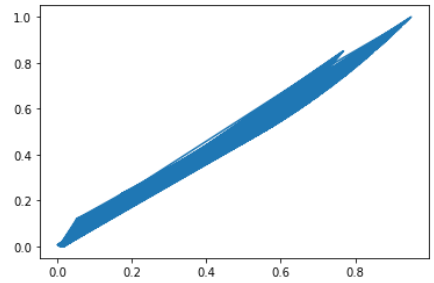


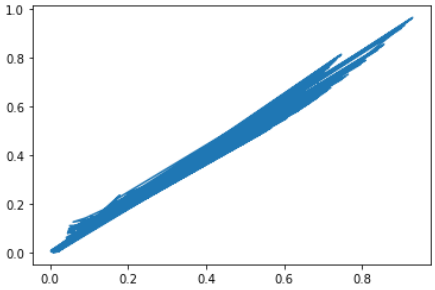
R2 for test & test predicted = **0.90**

Conclusion:- "adm" optimizer is giving the more r2 value as compare to "RMSProp" optimizer

Effect of changing the hidden layers:-

Number of hidden layers=4

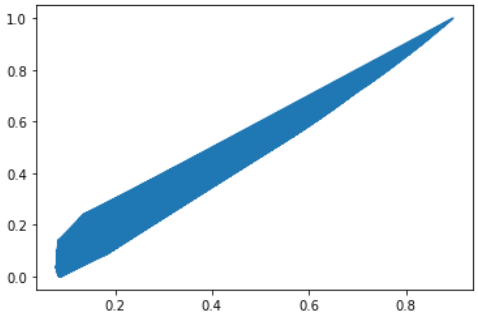


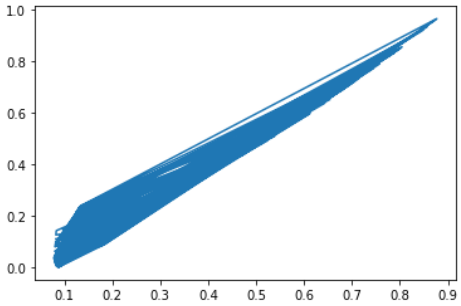


R2 for train & train predicted = **0.98**

R2 for test & test predicted = **0.98**

Number of hidden layers=3

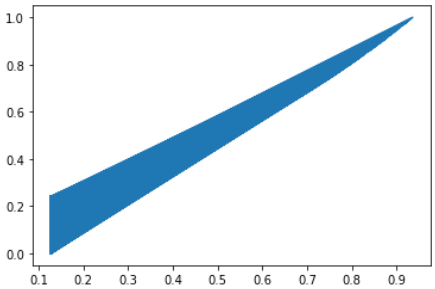


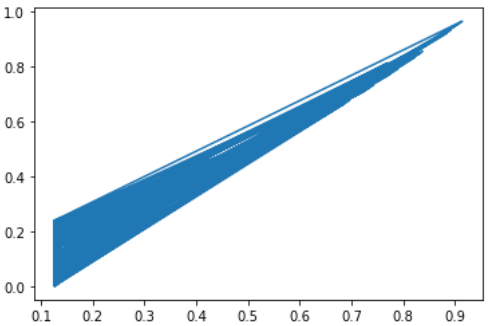


R2 for train & train predicted = **0.89**

R2 for test & test predicted = **0.91**

Number of hidden layers=2





R2 for train & train predicted =**0 .88**

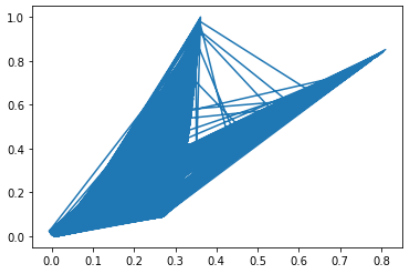
R2 for test & test predicted = **0.90**

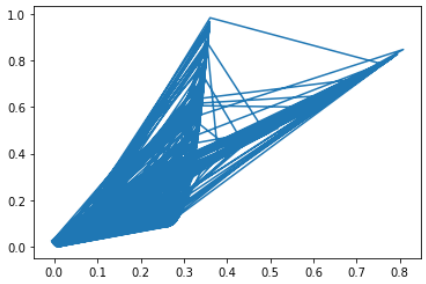
Conclusion: -

On increasing the number of hidden layers, the values of r2 are going to increase.

Effect of changing the sample size for training:-

Sample size=**60%**

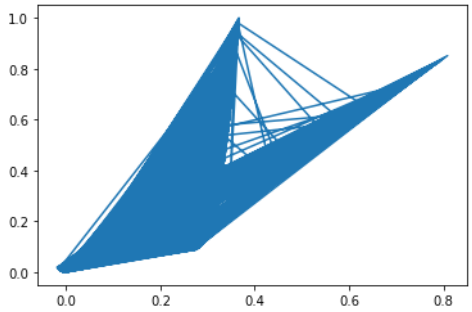


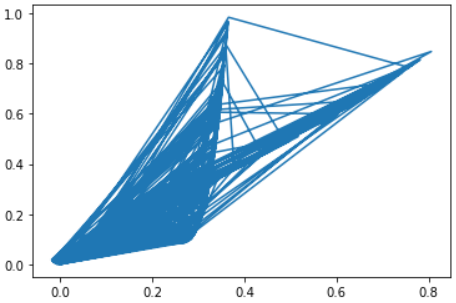


R2 for train & train predicted **=0 .55**

R2 for test & test predicted = **0.58**

Sample size = **65%**

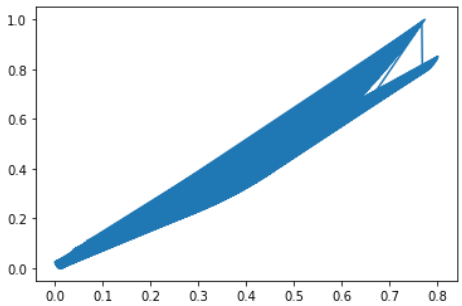


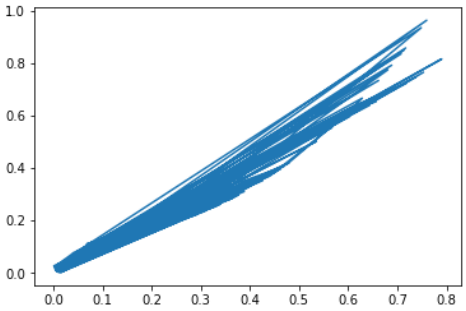


R2 for train & train predicted **=0 .56**

R2 for test & test predicted = **0.58**

Sample size = **75%**





R2 for train & train predicted **=0 .96**

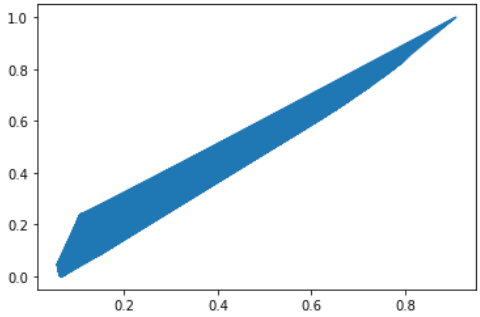
R2 for test & test predicted = **0.95**

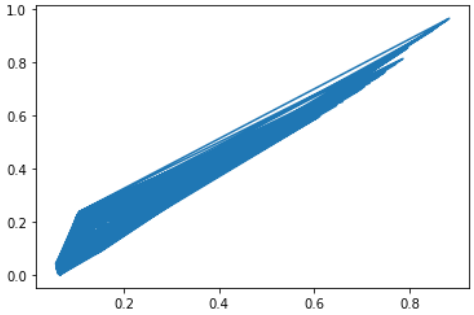
Conclusion: -

On increasing the sample size, the values of r2 are going to increase.

Effect of changing the epochs value :-

Epochs value=**100**

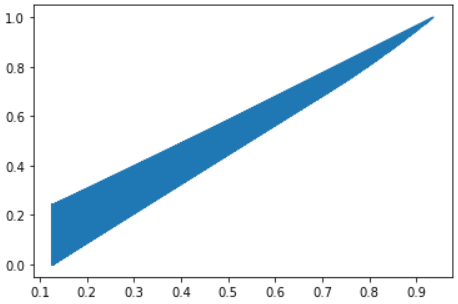


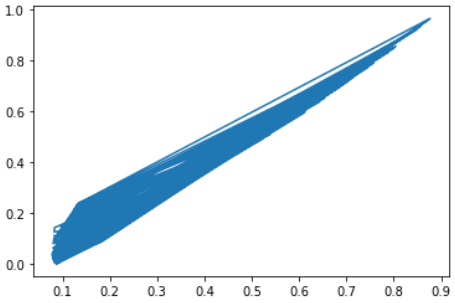


R2 for train & train predicted =**0 .88**

R2 for test & test predicted = **0.90**

Epochs value=**150**

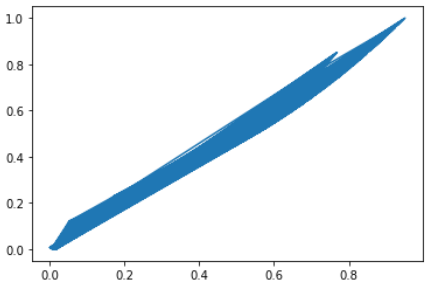


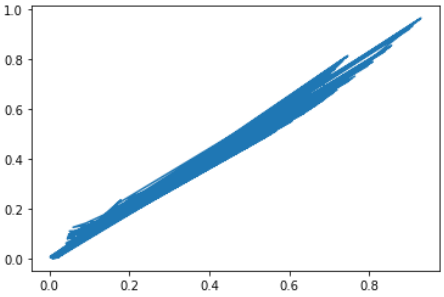


R2 for train & train predicted =**0 .89**

R2 for test & test predicted = **0.91**

Epochs value=**175**





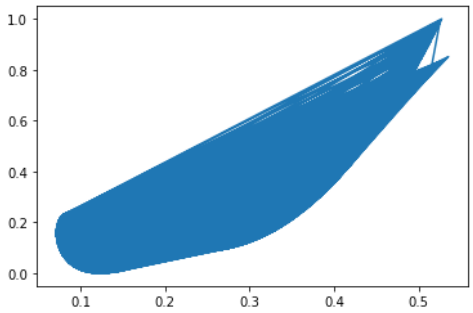
R2 for train & train predicted =**0 .89**

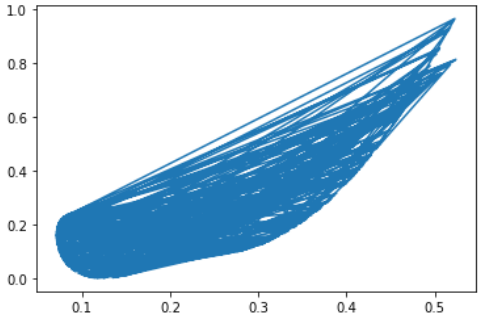
R2 for test & test predicted = **0.91**

Conclusion: - On increasing the epochs value, the values of r2 are going to increase.

Effect of changing the Activation function :-

Activation function: - **softmax**

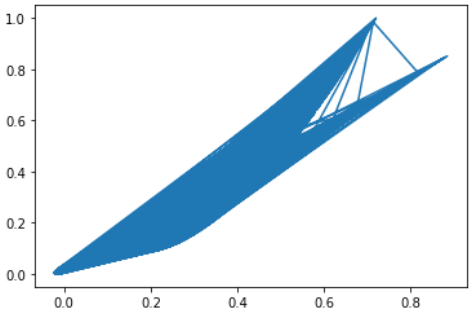


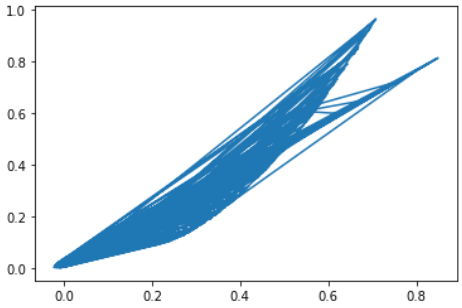


R2 for train & train predicted **=0 .54**

R2 for test & test predicted = **0.59**

Activation function: - **swish**

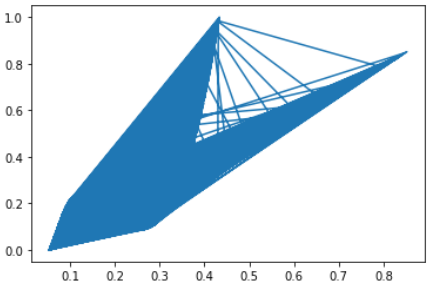


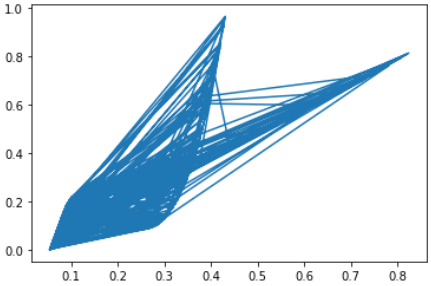


R2 for train & train predicted **=0 .54**

R2 for test & test predicted = **0.59**

Activation function: - **tanh**





R2 for train & train predicted **=0 .60**

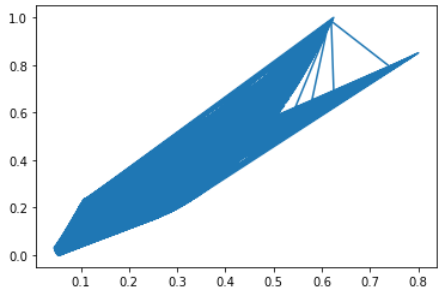
R2 for test & test predicted = **0.63**

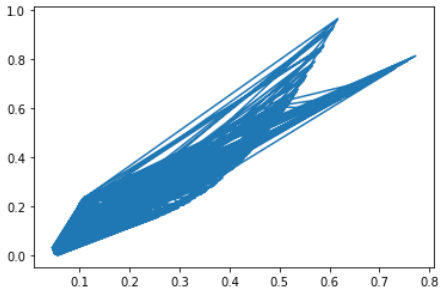
Conclusion: - On the activation function, the values of r2 are changing.

For different activation function, we are getting the different values of r2 .

Effect of changing the number of node :-

Hidden layers node:-(2,17,13,1)

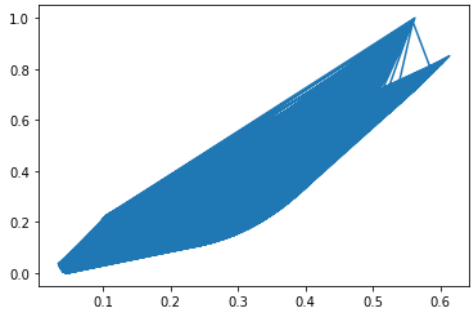


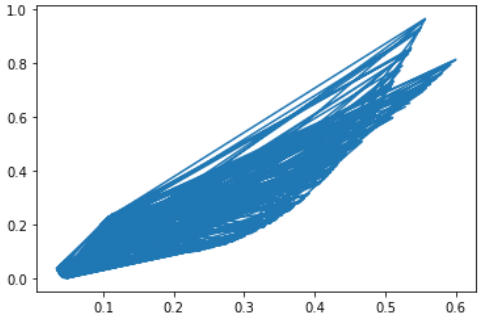


R2 for train & train predicted **=0 .80**

R2 for test & test predicted = **0.82**

Hidden layers node:-(2,19,15,1)

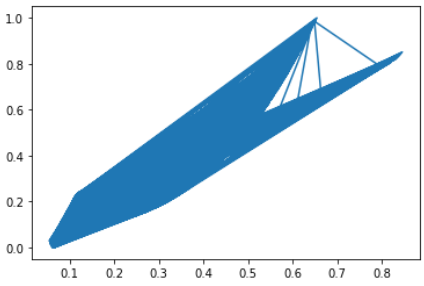


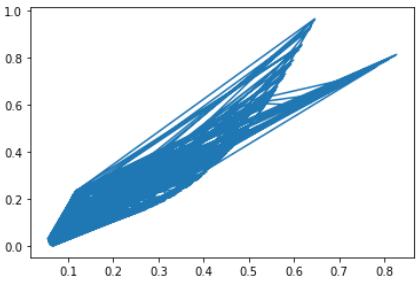


R2 for train & train predicted **=0 .73**

R2 for test & test predicted = **0.75**

Hidden layers node:-(2,20,16,1)





R2 for train & train predicted **=0 .82**

R2 for test & test predicted = **0.84**

Conclusion: -

On the hidden layers' node, the values of r2 is changing but there is no specific trend that on increasing the node value will increase or decrease.