ME5201: Computational Methods in Engineering

Report on Machine Learning Project

Submitted by

Thacker Setu Rameshbhai (ME23S027)

Department of Mechanical Engineering

Indian Institute of Technology Madras,

Chennai

Introduction to Project

Equation of 2D Sound wave propagation is given. Based on that to calculate Velocity of sound in medium it depends on Bulk modulus and Density of that medium. We have set of data for Bulk Modulus, Density and Velocity for that. We need to build ML model which can predict the value of Velocity based on Bulk Modulus and Density. We are going to build ML model using Linear Regression, Polynomial Regression of Degree 3 and Feedforward Neural Network.

Problem Statement

We have 1000 rows of data. Velocity of medium is depending on 2 Independent variable named Bulk Modulus and Density. We need to build a model for prediction of velocity based on given training data set.

Objective

Objective of the problem statement to solve 2D Wave equation using ML model not via Analytical method. ML model like, Linear Regression, Polynomial Regression and Feedforward Neural Network (FNN).

Approach for Problem Statement

We are using Pandas library for accessing the data in csv file. I have attached the screenshot of that how it is looking in IDE. I am using Jyputer Notebook as IDE for easy to access and handful for python Library. In this Project I am going to use multiple libraries like Numpy, Pandas, Scikit-Learn and many more like that.

```
[2]:              0             1        2
     0      density  bulk_modulus    speed
     1          818    1980000000  1449.28
     2         1600    1870000000   1010.1
     3         1830    2120000000  1002.51
     4         1780    1720000000   919.54
     …            …             …        …
     996       1900    1940000000   945.63
     997       1200    1060000000    883.0
     998       1820    2260000000  1041.67
     999       1660    1940000000   1010.1
     1000       488     939000000   1294.5

     [1001 rows x 3 columns]
```

Let us understand why we use these Libraries:

1. Numpy: NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.
2. Pandas: Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data.
3. Scikit-Learn: Scikit-learn is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction. We are using Scikit-Learn library for its regression purpose.

We have solved this problem with Three different kind of ML Algorithms. Let us understand basics of these algorithms one by one.

1. Linear Regression

$$y = w_0 + w_1 x_1 + w_2 x_2$$

Where, $w0$, $w1$, $w2$: Weights

$x1$ = Bulk Modulus (First Independent variable)
$x2$ = Density (Second Independent variables)
y = velocity (Dependent variable)

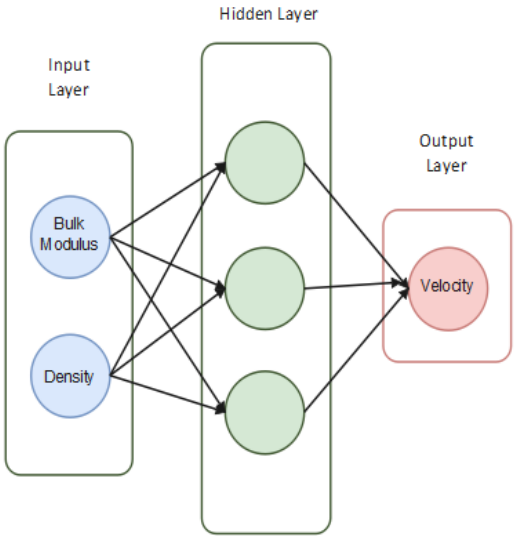
2. Cubic Regression

$$y = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1 x_2 + w_4 x_1^2 + w_5 x_2^2 + w_6 x_1^2 x_2 + w_7 x_2^2 x_1 + w_8 x_1^3 + w_9 x_2^3$$

Where $w0$, $w1$, $w2$ … … … … …. $w9$*: weights*

3. Feedforward Neural Network

A feedforward neural network is one of the simplest types of artificial neural networks devised. In this network, the information moves in only one direction—forward—from the input nodes, through the hidden nodes (if any), and to the output nodes. There are no cycles or loops in the network. It has 2 neurons in Input Layer, 1000 neurons in 1 Hidden Layer and 1 neuron in output layer. We need to use on 1 hidden layer only as per the problem statement.



Model fitting for Linear Regression

These are the steps which I perform for getting the result using Linear Regression:

1. Import all the important libraries which we are going to use
2. Fetch the data from CSV input file which contain 1000 rows of data
3. Define X and y. X represent Input variable and y represents Output variable
4. By using train_test_split method from Scikit-Learn library, we divide our data into 8:2. 80% data should be use for Training Purpose and 205 data should be used for Testing purpose

```
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,␣
 ↪random_state=42)
print(len(X_train),len(X_test))
```

```
800 200
```

5. MinMaxScalar is library which is useful when data have some irregularities in it. In our case data of Density is in terms of 10^3 and Bulk Modulus is in order of 107. So, there should be applied normalization to scaling down. It also preserv the distribution after scaling down.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

6. We feed the 80% of data to Linear regression model for training purpose. And based on that we derive some matrices for checking the performance of the model which we have built using limited data.

Mean squared Error	Mean absolute Error	R2 score	Data
19581.5131619656	65.93248602236183	0.763621774060595	Training

Mean squared Error	Mean absolute Error	R2 score	Data
5428.169870187041	56.75077145048133	0.918258157223190	Testing

It looks, that there are some spaces to improve. Let's try with Cubic Regression.

Model fitting for Polynomial Regression (Degree = 3)

I have done same step to do polynomial regression as linear regression. Here is the performance parameter for Polynomial Regression

Mean squared Error	Mean absolute Error	R2 score	Data
10623.40294532145	31.28541527703838	0.871759596876707	Training

Mean squared Error	Mean absolute Error	R2 score	Data
1238.776912571769	21.96454555512266	0.981345479223277	Testing

Model fitting for Feedforward Neural Network (FNN)

We have imported MLPRegressor from Skicit-Learn library and used 1000 neurons in 1 hidden layer, Logistic function as activation function, LBFGS as Solver and Maximum iteration about 2000. These values I have found out based on multiple permutation and combination. I choose these values of parameter because it gives highest accuracy among all and it took less time for computation compare to others. After that I fit the data into model and get the performance parameter.

```
nn = MLPRegressor(hidden_layer_sizes = (1000), activation= 'logistic',␣
 ↪max_iter= 2000, solver= 'lbfgs')
nn.fit(X_train, y_train)
```

Mean squared Error	Mean absolute Error	R2 score	Data
8.286338307879548	1.769148634931958	0.999899971471430	Training

Mean squared Error	Mean absolute Error	R2 score	Data
145.1801103391561	4.348751996044686	0.997813758589457	Testing

Conclusion

As we can see that, Feedforward Neural network has outperformed all other ML Algorithms like Linear and Polynomial Regression. We have decided our best ML Algorithm based on performance parameters like Mean Squared Error, Mean absolute Error and R2 Score.

Let us compare these performance parameters on Test data and choose the best ML Algorithm:

Performance Parameter	*Linear Regression*	*Polynomial Regression*	*Feedforward Neural Network (FNN)*
Mean Squared Error	5428.1698	1238.7769	145.1801
Mean Absolute Error	56.7508	21.9645	4.3488
R2 Score	0.9182	0.9813	0.9978

We conclude that, based on table FNN model have highest accuracy or less error in its prediction compare to other models.