

Predicting Strength of High-Performance Concrete Using Artificial Neural Network

by

Mayank Pathania

Roll No. - 204103314

Specialization - Machine Design



Indian Institute of Technology, Guwahati
Guwahati - 781039, India

April 8, 2021

Contents

| | |
|---|-----------|
| 1. Introduction | 2 |
| 1.1 Problem Definition | 2 |
| 1.2 Attribute Information | 2 |
| 2. Methodology | 3 |
| 2.1 Analysing Data | 3 |
| 2.2 Model Architecture | 5 |
| 2.3 Model Parameters | 5 |
| 3. Results and Discussions | 8 |
| 3.1 Convergence Plots | 8 |
| 3.2 Predictions | 10 |
| 3.3 Comparison | 11 |
| 3.3.1 Convergence Plots | 11 |
| 3.3.2 Predictions | 12 |
| 4. Conclusion | 13 |
| 4.1 Some Remarks Regarding Code | 13 |
| References | 13 |

Introduction

1.1 Problem Definition

Concrete is the most important material in civil engineering. The concrete compressive strength is a highly nonlinear function of age and ingredients.

The objective of the problem is to predict the strength of High-performance concrete based on some parameters. There are mainly 8 parameters Cement, Blast Furnace Slag, Fly Ash, Water, Super-plasticizer, Coarse Aggregate, Fine Aggregate and Age that effect the strength of the High-performance concrete. The mathematical modeling of such system is not easy and is highly non-linear. So a soft computing technique, Artificial Neural Network, is used to test if it is possible to predict the Strength using Artificial Neural Network and given parameters.

1.2 Attribute Information

The data is taken from UCI Machine Learning Repository.

In table 1.1 the name for the component, type of data, Measurement units and description whether input or output variable is given. The order of the components is in table same as ordered in the data files.

| Name | Data Type | Measurement | Description |
|-------------------------------|--------------|--------------------|-----------------|
| Cement | quantitative | kg in a m3 mixture | Input Variable |
| Blast Furnace Slag | quantitative | kg in a m3 mixture | Input Variable |
| Fly Ash | quantitative | kg in a m3 mixture | Input Variable |
| Water | quantitative | kg in a m3 mixture | Input Variable |
| Superplasticizer | quantitative | kg in a m3 mixture | Input Variable |
| Coarse Aggregate | quantitative | kg in a m3 mixture | Input Variable |
| Fine Aggregate | quantitative | kg in a m3 mixture | Input Variable |
| Age | quantitative | Day (1–365) | Input Variable |
| Concrete compressive strength | quantitative | MPa | Output Variable |

Table 1.1: Information of components of data

Methodology

2.1 Analysing Data

Scatter plots of all the input parameters are plotted with the output parameter(Strength).

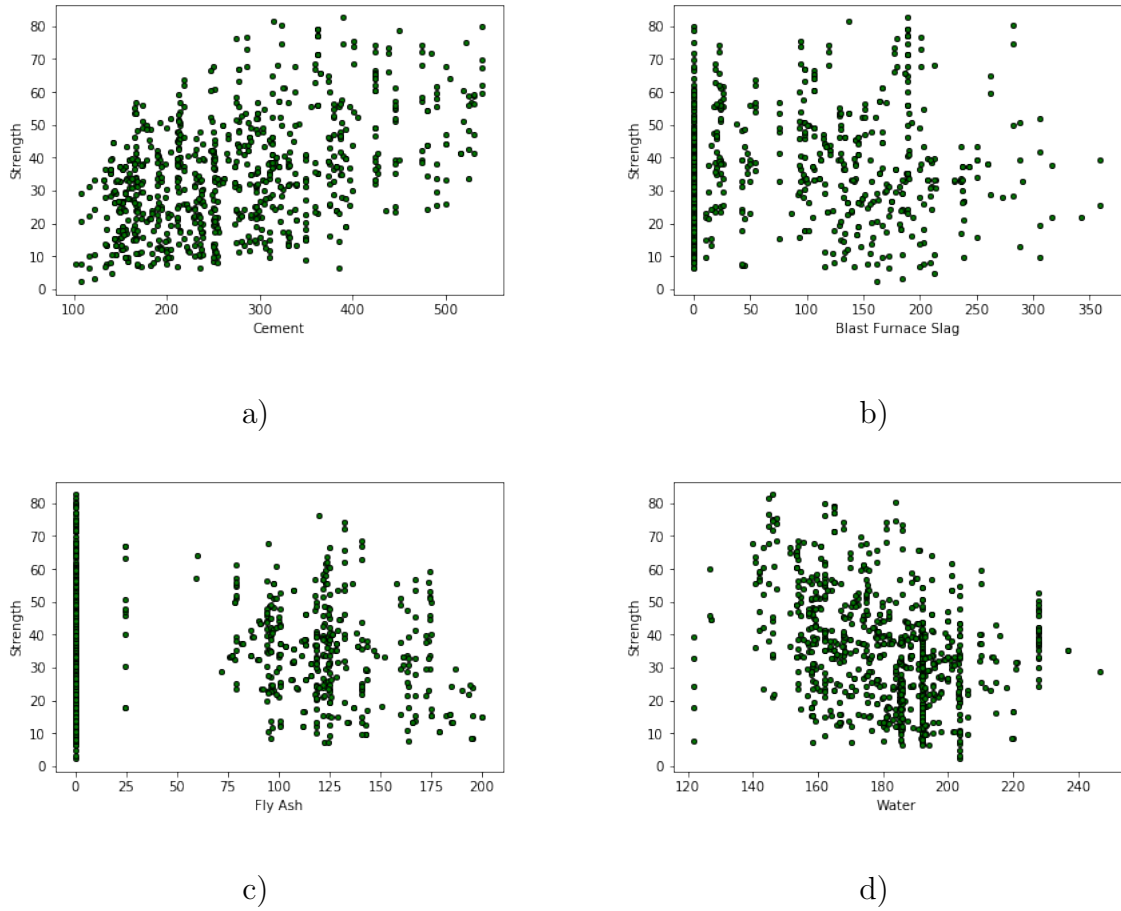


Figure 2.1: Scatter the plots between Strength and a) cement; b) Blast Furnace Slag; c) Fly Ash; d) Water;

It can be noted that the data is real valued except for Age which is integer valued and it is hard to visualize any linear relationship between input parameters and strength. As it is already mentioned that the relation between strength and input parameters is highly non-linear the a graphs are as expected.

For the Cement it can be seen that with the increase of value of cement there is increase in strength but it can also be noted that the strength is highly affected by other parameters as even for high cement values, around $500kgm^{-3}$, strength value varies low to high, around $40MPa$ to $80MPa$.

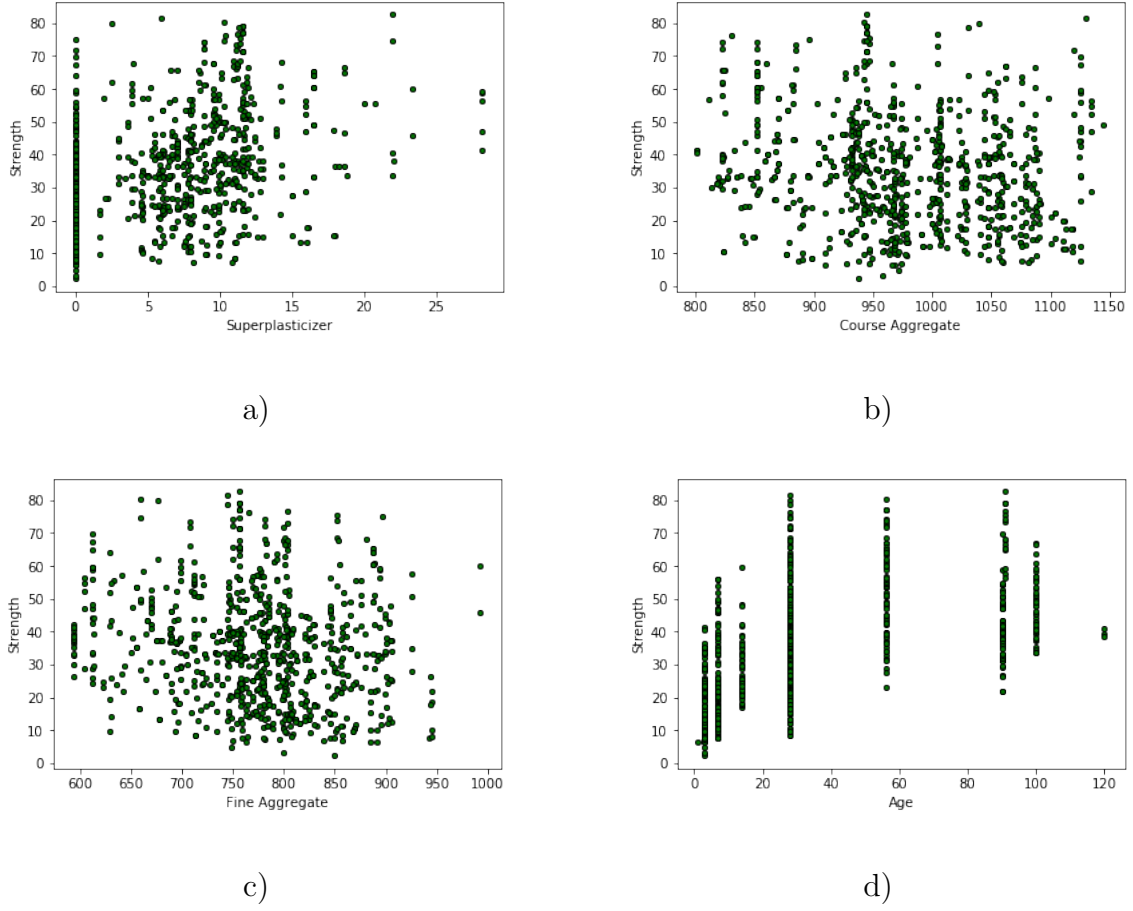


Figure 2.2: Scatter plots between strength and a) Superplasticizer; b) Coarse Aggregate; c) Fine Aggregate; d) Age

Now the values of data are presented in table to see the maximum, minimum, mean and median of the data in their respective units as discussed in Table 1.1. This tells us the need of normalization of the data as values are not varying in same range and this can lead to biases in prediction. For example cement is ranging from 102 to 540 kgm^{-3} and superplasticizer is ranging from 0 to 22.1 kgm^{-3} this may lead to biases toward cement values.

| <i>attribute</i> | <i>minimum</i> | <i>maximum</i> | <i>Average</i> | <i>median</i> |
|-------------------------|----------------|----------------|----------------|---------------|
| <i>Cement</i> | 102.0 | 540.0 | 277.093 | 257.75 |
| <i>BlastFurnaceSlag</i> | 0.0 | 342.1 | 76.9958 | 22.0 |
| <i>FlyAsh</i> | 0.0 | 195.0 | 57.6307 | 0.0 |
| <i>Water</i> | 127.0 | 228.0 | 180.1824 | 183.0 |
| <i>Superplasticizer</i> | 0.0 | 22.1 | 6.4395 | 7.0 |
| <i>CoarseAggregate</i> | 801.0 | 1145.0 | 976.1921 | 972.2 |
| <i>FineAggregate</i> | 594.0 | 945.0 | 774.1864 | 778.25 |
| <i>Age</i> | 1.0 | 120.0 | 32.3507 | 28.0 |
| <i>Strength</i> | 2.33 | 79.4 | 35.415 | 33.724 |

Table 2.1: Aspect of components of data

2.2 Model Architecture

The Artificial Neural Network model with one input layer, one hidden layer, and one output layer was considered for the analysis. Model is fully connected network with batch mode of training. The number of input and output neurons is 8 and 1 respectively and decided by the number of input and output parameters. Number of hidden neurons can be varied.

As the output parameter is Strength which is real valued transfer function for both hidden

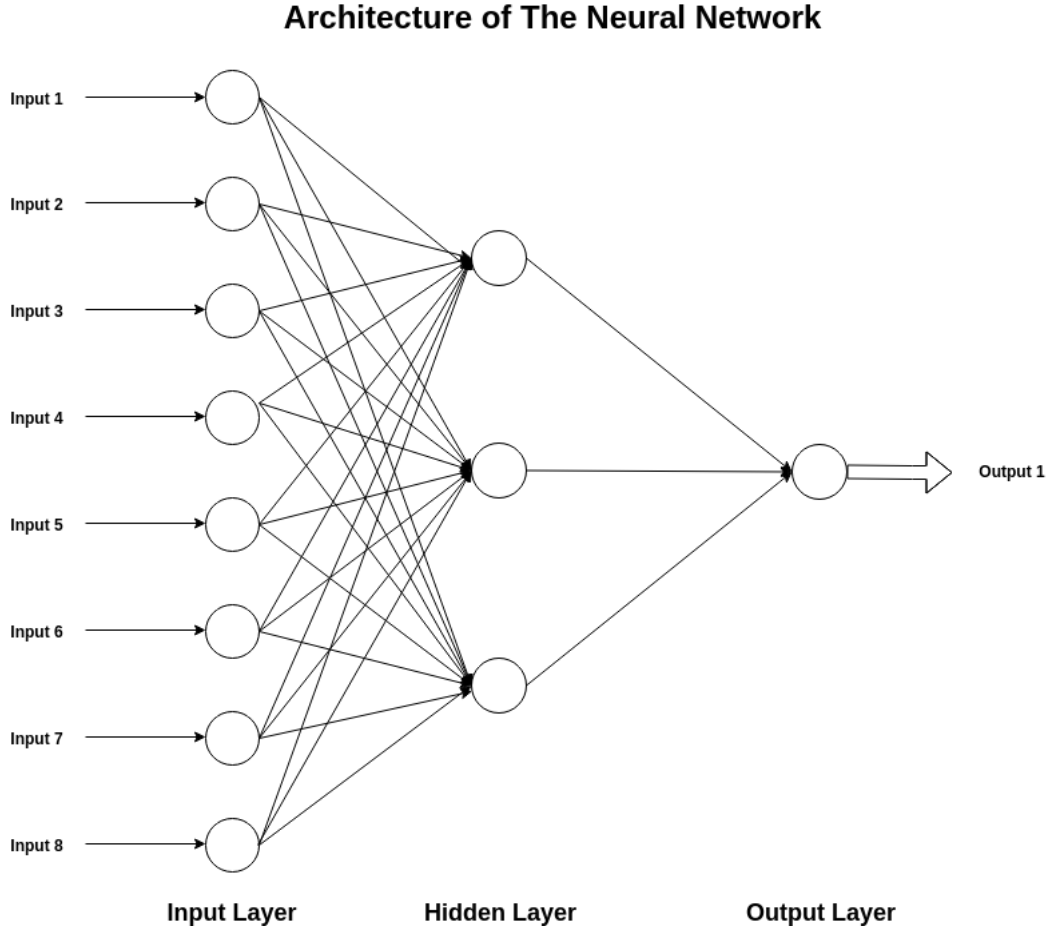


Figure 2.3: Architecture of neural network

and output layer was selected as log sigmoid transfer function with $a = 1$. The values of input parameters were normalized within range -4 to 4 for all parameters before feeding into the network. Also the target values are normalized in the range 0.1 to 0.9.

$$y = \frac{1}{1 + e^{-ax}}$$

Figure 2.4: Log sigmoid transfer function

2.3 Model Parameters

Number of hidden neurons, learning rate, momentum coefficient and tolerance value are the model parameters which can be changed to alter the convergence rate and termination

criterion of the training. These values were chosen by trial and error method. Different sets of data was feed into the neural network with different model parameters and then number of iterations were noted for a given torelance value. Model parameters which gave the best results i.e. minimum number of iterations were chosen as final parameters.

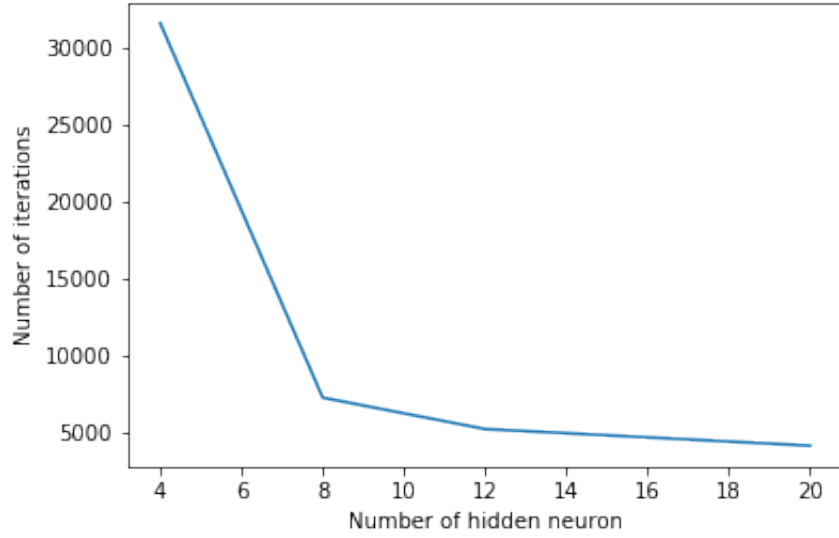


Figure 2.5: Comparison of number of iterations for tolerance = 10^{-5} , learning rate = 0.3 and momentum coefficient = 0

First the value of learning rate and momentum coefficients were fixed and for hidden neurons 4, 8, 12 and 20 the training was performed and the number of interation taken for training were noted.

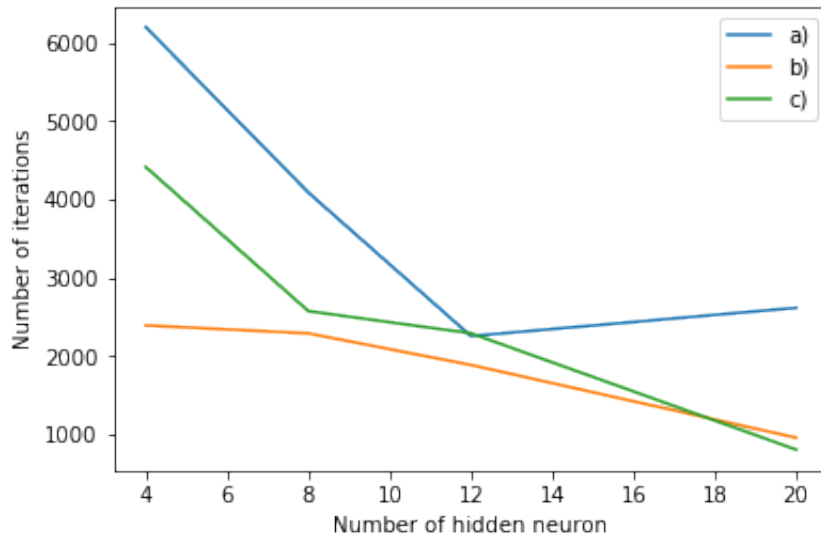


Figure 2.6: Comparison of number of iterations for tolerance = 10^{-5} and
a) learning rate = 0.3 and momentum coefficient = 0
b) learning rate = 0.7 and momentum coefficient = 0.5
c) learning rate = 0.9 and momentum coefficient = 0.6

Then the values for learning rate and momentum was increased and for different number of hidden neurons again the number of iterations were noted. This process was repeated for various values of model parameters. For this procedure the value of tolerance was taken as 10^{-5} but after the selection of parameters the value of tolerance was further decreased to 5×10^{-6} as this increased the accuracy in prediction and increase in number of iterations was not much. But further reduction in tolerance value increased the number of iteration very much.

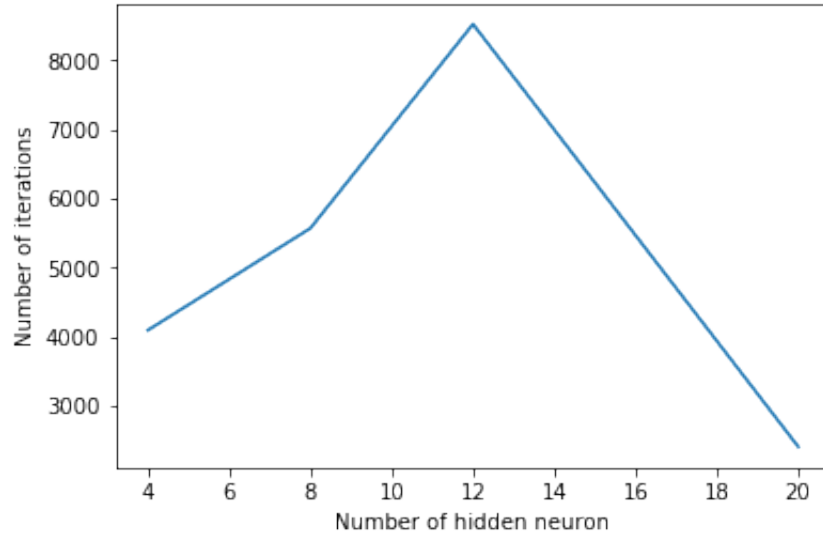


Figure 2.7: Comparison of number of iterations for tolerance = 5×10^{-6} , learning rate = 0.9 and momentum coefficient = 0.6

For tolerance, it was noted that for tolerance value of 10^{-6} the results of training and testing were good and by further reduction in its value increased the number of iteration but there was not much improvement in predictions. For instance, with tolerance = 10^{-7} it took 10 to 20 times more iterations to converge than with tolerance value of 10^{-6} , but the Mean Square Error for testing in both the cases was of order 10^{-6} .

| Model Parameter | Value |
|-----------------------|--------------------|
| No. of Hidden Neurons | 20 |
| Learining Rate | 0.9 |
| Momentum Coefficients | 0.6 |
| Error/Tolerance | 5×10^{-6} |

Table 2.2: Decided Values of Parameters

Results and Discussions

3.1 Convergence Plots

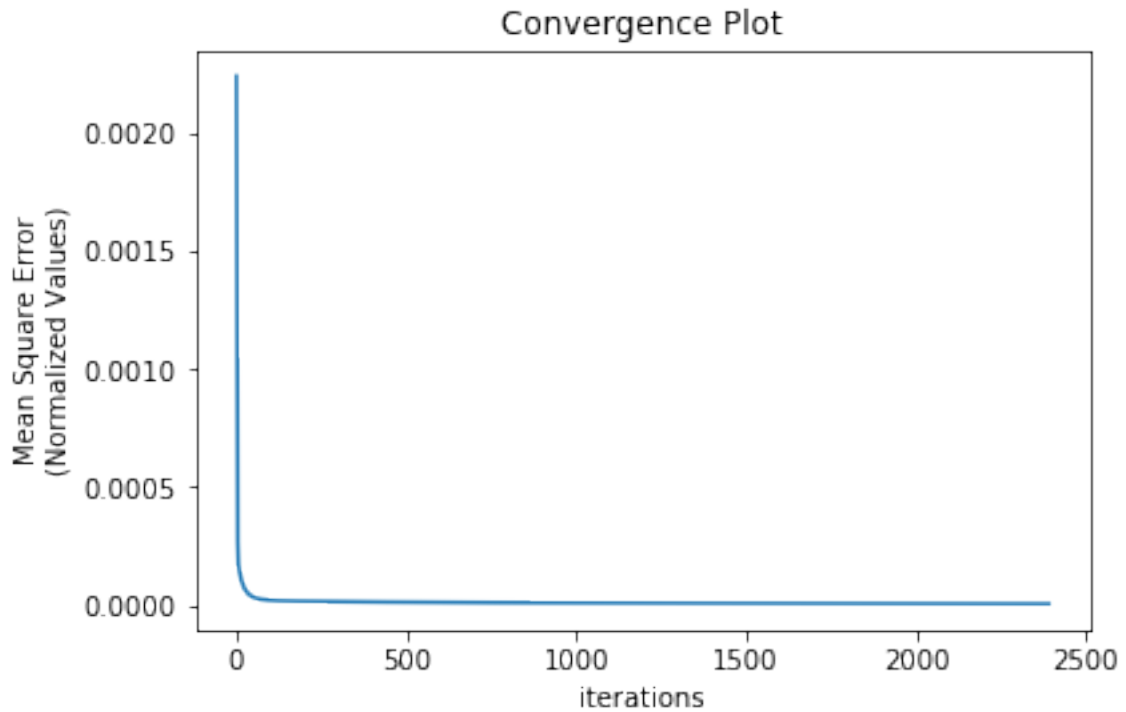


Figure 3.1: Convergence plot

The convergence plot is for the selected parameters i.e. No. of hidden neurons = 20, Learning Rate = 0.9, Momentum Coefficients = 0.6 and error/Tolerance = 5×10^{-6} . From the convergence plot it can be seen that for the first few iterations, around 5-10, there is a major drop in the Mean Square Error (MSE) and after that the curve seems to be flat. To better visualize how the error is reducing after these iterations, two more convergence plots are shown in figure 3.2 and figure 3.3.

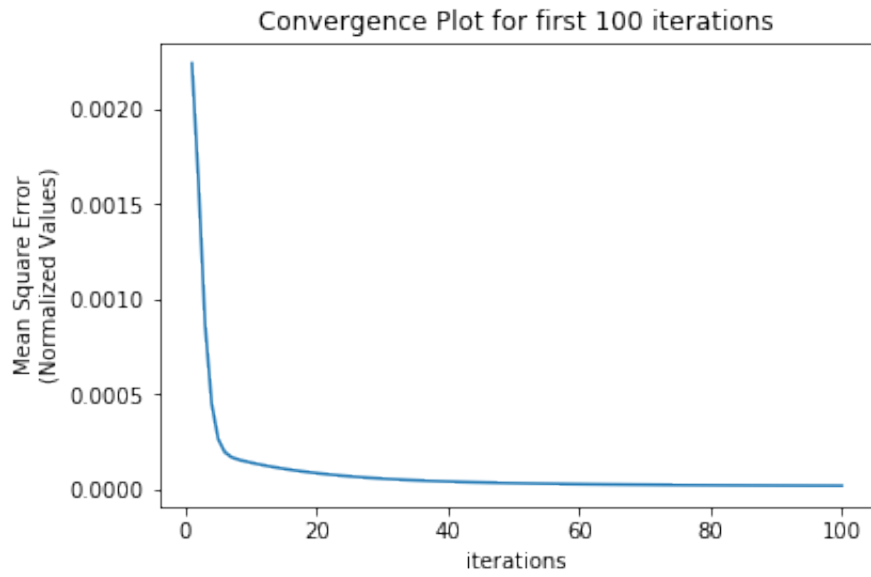


Figure 3.2: Convergence Plot for first 100 iterations

From figure 3.2 it can be seen that the reduction in error is very high in first few iterations and around 20 iterations it seems as if the curve is flattening. As the weight values at the start are random the error is high and as the training is proceeding the error is reducing.

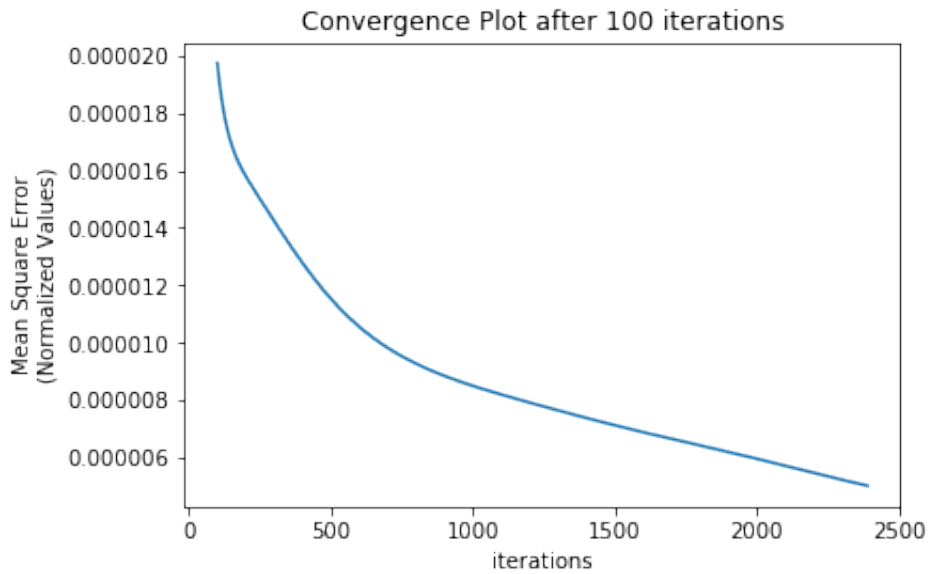


Figure 3.3: Convergence Plot after 100 iterations

From figure 3.3 we can appreciate how the error is reducing continuously and the curve actually do not go flat. Even at termination condition curve is quite steep and with further reduction in tolerance value of termination the MSE will further reduce.

3.2 Predictions

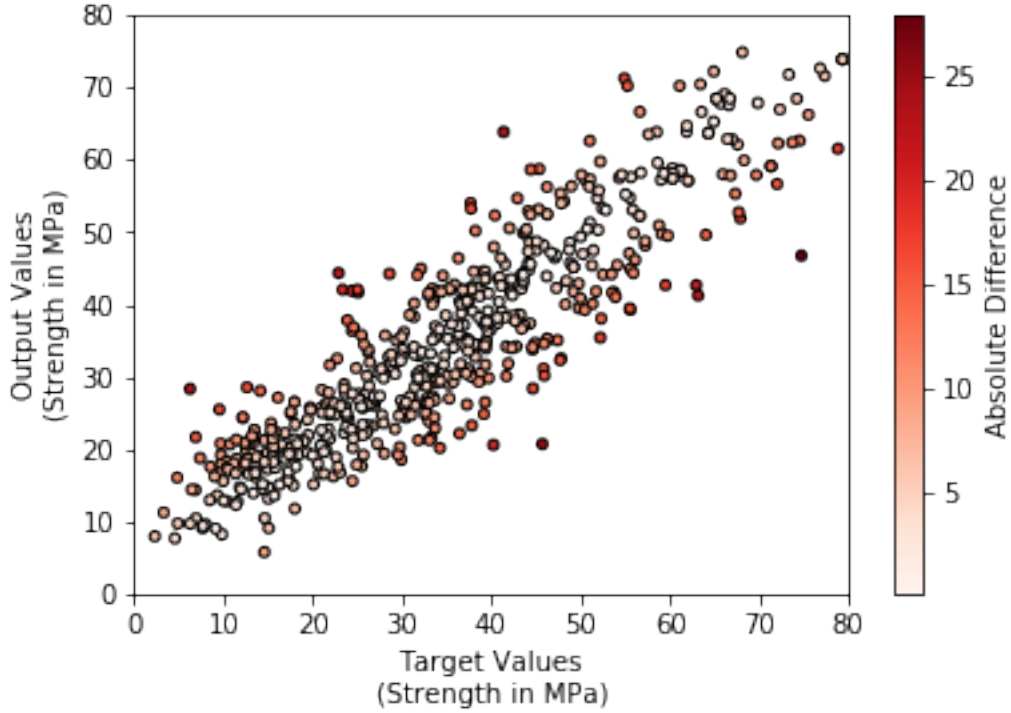


Figure 3.4: Comparison of Target values and Output Values

Plot in figure 3.4 shows that the prediction values are close to the target values. Variation of Absolute difference between predicted and target values is same throughout. So it cannot be concluded if the method is better for low strength or high strength high-performance concrete. Also the model is Underpredicting and overpredicting by same amount as it can be see by the color which is similar on both sides of portion of the graph with white points.

| | |
|------------------------|---------|
| Maximum absolute Error | 27.9126 |
| Minimum absolute Error | 0.0155 |
| Mean absolute Error | 5.7182 |
| Mean square Error | 52.9724 |

Table 3.1: Errors in Training Ouput

| | |
|------------------------|---------|
| Maximum absolute Error | 22.3789 |
| Minimum absolute Error | 0.01622 |
| Mean absolute Error | 5.3483 |
| Mean square Error | 48.7034 |

Table 3.2: Errors in Testing Ouput

Tables 3.1 and 3.2 shows various errors in prediction by the ANN Model for training and testing respectively. For both cases the values of errors are almost similar indicating the model has not over-fitted the training data. The maximum absolute error in both cases is high.

3.3 Comparison

In this section a comparison is made between the Model with selected parameters and a model with same parameters and lower tolerance i.e. 10^{-6} . Though the reduction is only of 5×10^{-6} it will lead to some interesting results.

3.3.1 Convergence Plots

Similar to the section 3.1 here 3 convergence plots are shown in figure 3.5.

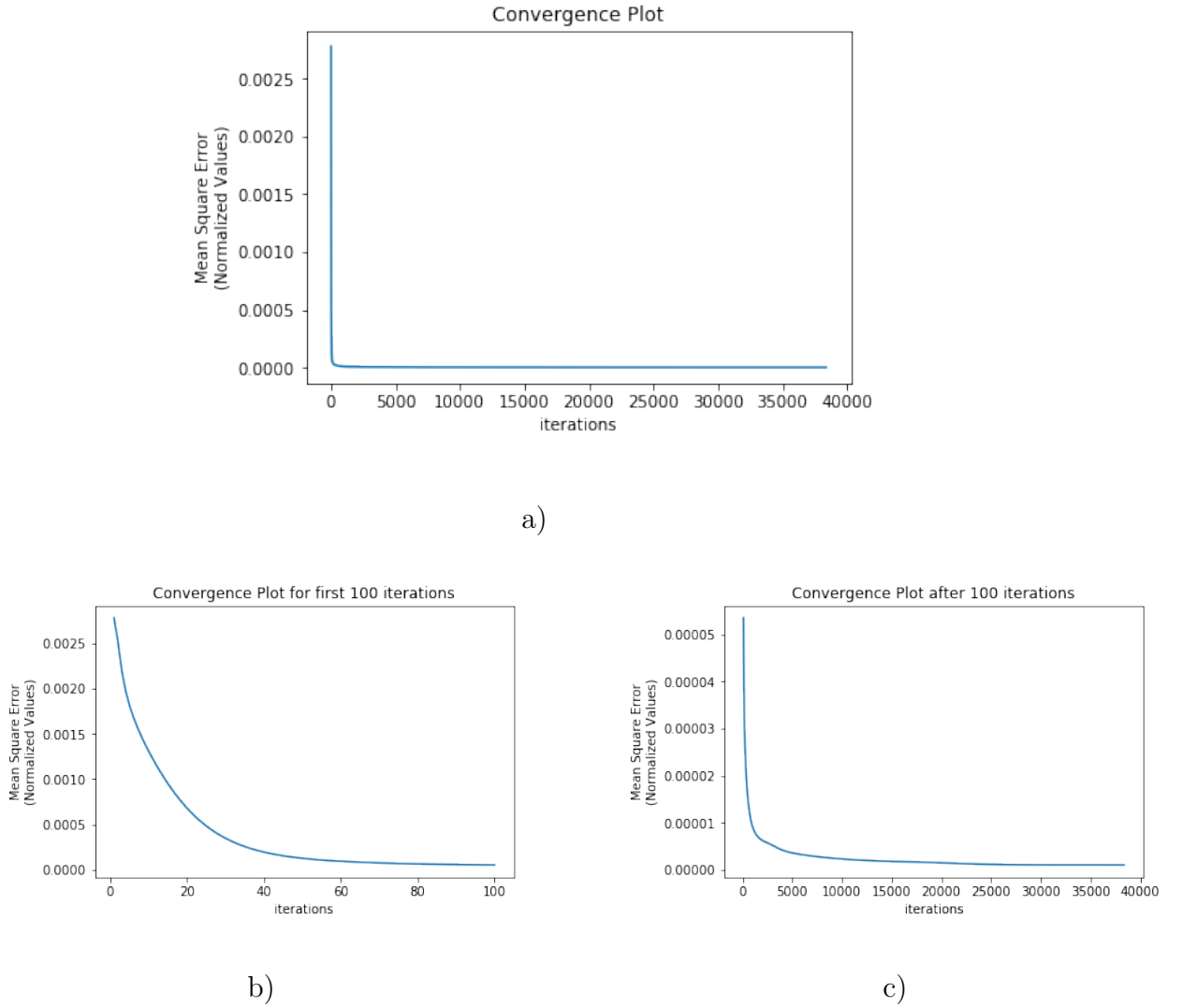


Figure 3.5: Convergence Plot for a) All iteration; b) First 100 iterations; c) After 100 iterations

In figure 3.4 a) similar trend can be seen in the plot that were seen in figure 3.1. The convergence rate is high in the starting and then it is low as the error is reduced to around 4×10^{-5} . From figure 3.5 b) for first 100 iterations trend is also same that the curve is flattening as the number of iterations are increasing. After 100 iterations the trend in starting of figure 3.5 c) is similar to figure 3.3 but there is a change in the later stage. At 5,000 iterations the error is 3×10^{-6} and after that there is not much change in error and convergence rate

is very low and it takes total of 38373 iterations for termination criterion to be fulfilled i.e. $MSE < 10^{-6}$. The curve is almost flat in this case while for that in figure 3.3 the curve was quite steep.

3.3.2 Predictions

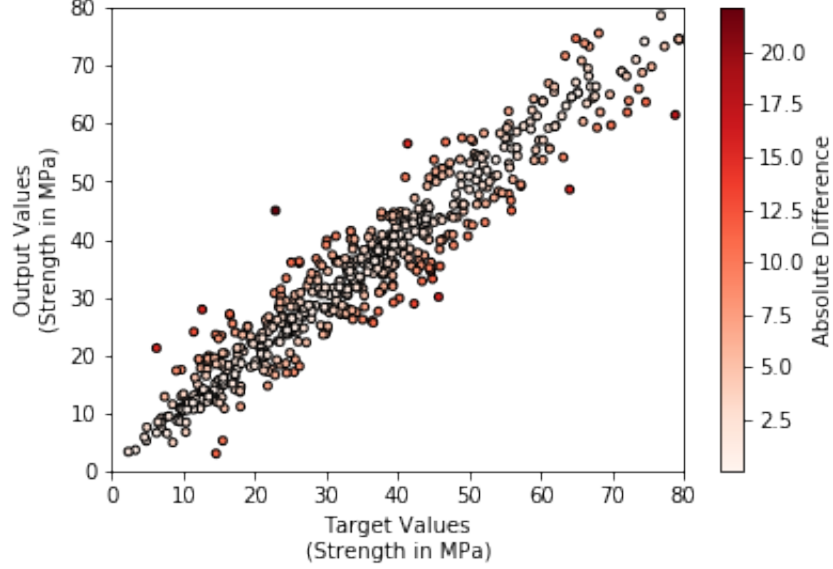


Figure 3.6: Comparison of Target values and Output Values

Plot in figure 3.6 shows that the prediction values are close to the target values. Points are more close to $y = x$ line than in figure 3.4 and also the colors are not as dark for the values. This implies that most of the predictions are close to the targets and there are lesser number of values which are far from target. This can also be seen from the tables 3.3 and 3.4 that there is a reduction in Mean Square Error and Mean Absolute Error.

| | |
|------------------------|----------|
| Maximum absolute Error | 22.11 |
| Minimum absolute Error | 0.005 |
| Mean absolute Error | 3.7328 |
| Mean square Error | 23.69096 |

Table 3.3: Errors in Training Ouput

| | |
|------------------------|---------|
| Maximum absolute Error | 24.924 |
| Minimum absolute Error | 0.0625 |
| Mean absolute Error | 4.3087 |
| Mean square Error | 32.4043 |

Table 3.4: Errors in Testing Ouput

Conclusion

High-performance concrete is highly complex material and its strength depends on various parameters. 8 parameters were considered and different model parameters for the ANN Model were used to analyse which will give better results and if it is possible to use this technique to predict high-performance concrete.

Following Conclusions can be drawn from the study:

- This Method can be used to predict the strength of high-performance concrete also ANN model is simple to implement but it will require data and computation resources to train the model upto required accuracy.
- Changing the parameters gave different convergence rates. Learning rate = 0.9 was used which is high value but it reduced the number of iterations significantly and did not produced any instability.
- Reducing the tolerance value too much might not be helpful as MSE for training will decrease but this might be due to overfitting and during the testing MSE might be only of order 10^{-6} . More data for training may be helpful but this will also increase the computational time/resources.

4.1 Some Remarks Regarding Code

- Code is generalized. Input and output parameters can be changes from the file "input_parameters.dat".
- Model parameters can be changed by changing values in the file "model_parameters.dat". There is only single hidden layer and cannot be changed but number of hidden neurons can be changed.
- After training the model for reuse of model "save_model_data()" function can be used to save the data and "load_model_data()" function can be used to load the saved data.

References

1. I-Cheng Yeh, "Modeling of strength of high performance concrete using artificial neural networks," Cement and Concrete Research, Vol. 28, No. 12, pp. 1797-1808 (1998).