Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 3159-0040
Vol. 2 Issue 5, May - 2015

Neural Network Modeling of Concrete Slump and Compressive Strength of a Nigerian Portland Cement

Zakka Ze Gyang,
Department of Building,
Abubakar Tafawa Balewa University,
Bauchi, Nigeria.
zezakka@gmail.com

Zakka Benisemeni Ze,
Computer Science Department,
Federal Polytechnic,
Bauchi, Nigeria.
byikawe@yahoo.com

Muhammad Nura Isa
Department of Building,
Abubakar Tafawa Balewa University,
Bauchi, Nigeria.
mni211@yahoo.com

Abstract—Concrete mix designs are usually succeeded by trial mixes which allow for adjustments to the mix constituents; in most instances this is time consuming and costly. This study is aimed at presenting a model for determining the slump and 28 day compressive strength of fresh and hardened concrete specimens respectively of a Nigerian Portland cement of grade 32.5. The DOE method of mix design was used in predetermining the ratio of materials to be mixed. 36 mix designs were done with 6 cubes casted per each mix (216 concrete cubes in all) of size 100×100×100mm were used in developing the Artificial Neural Network model. The slump of the mix was determined before it was casted. 60% of the samples were randomly selected and used for the training of the Feed-Forward Neural Network while 20% each were used for the testing and validation of the model. The training, testing and validation of the Neural Network were successful. 5 neurons (water content, cement content, coarse aggregate size, coarse aggregate weight and fine aggregate weight) were used in the input layer, 10 and 8 neurons were used in the first and second hidden layers while 2 neurons (slump and compressive strength) were used in the output layer. A learning rate of 0.0001 was used. The network trained after 136 epochs with a Mean Square Error of 0.0134525.

Keywords: Concrete Mix, Artificial Neural Network, Training, Neurons, Layers.

1. Background

The Nigerian construction industry witness frequent building collapses. This trend has been the foremost concern of the construction industry professionals with the view to curb this trend. Some key factors such as poor supervision, substandard building materials and use of inexperienced workers among others have been identified as the cause of building collapses [1]. In the materials, concrete is most prone to inconsistency when its production is not well supervised.

To achieve concrete of desired quality, a mix design which is defined as the process of selecting suitable ingredients of concrete and determining their relative quantities with the objective of producing, as economic as possible, concrete of certain minimum properties, notably strength, durability, and a required consistency is worked out after which trial mixes are conducted [2]. This process is time consuming, leads to wastage of concrete materials thereby inadvertently increasing cost. It is in view of these shortcomings that this model which employs Artificial Neural Network (ANN) was designed for a selected Portland cement brand.

The ANN has the ability to learn complex relationships between the inputs and target results and if trained can predict future outcomes when given new data. This paper also intends to provide data which other construction industry workers can use to further broaden the use of this model as its efficiency depends on the quantity if data provided.

Neural Network

The Neural Network (NN) is a toolbox in the MATLAB software. It is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include;

- Math Computation,
- Algorithm Development,
- Data Acquisition,
- Modeling, Simulation, and Prototyping,
- Data analysis, Exploration, and Visualizations,
- Scientific and Engineering Graphics,
- Application development, including graphical user interface.

The NN toolbox also includes many kinds of powerful networks for solving problems including;

- Function approximation, modeling,
- Signal processing and prediction,
- Classification and clustering.

These tools are an essential part of many applications, including engineering, finance, medicine, and artificial intelligence (AI).

Artificial Neural Networks (ANN)

Artificial Neural Networks (ANN) was developed to model the human brain. Even a fairly simple and small sized ANN, when compared to the human brain, has some powerful characteristics in knowledge and information processing due to its similarity (to the brain). ANNs can 'learn' complex input-output mappings. The mappings are not specified, but 'learned'. The learning process is used to determine proper interconnection weights, and the network is 'trained' to properly associate the inputs with their corresponding outputs.

An ANN has been described by Tazelaar, [3] as "humanity's attempt to mimic the way the brain does things in order to harness its versatility and ability to infer and intuit from incomplete or confusing information". More specifically, they can be used to learn complex relationships for the recognition of patterns.

An ANN, described in detail by Zurada [4], is first 'trained' to identify the relationship between a series of input vectors and their corresponding output vector. Input vectors are repeatedly presented to the network one at a time, each element of the vector corresponding to a different neuron in the first layer of the network (see Fig. 1)

These inputs are then multiplied by a corresponding weight before being forwarded to every neuron in the next layer of the network. The forwarded values are multiplied by the weights associated with each neuron (see Fig. 2) before being summed together with all other values and sent simultaneously to that neuron.

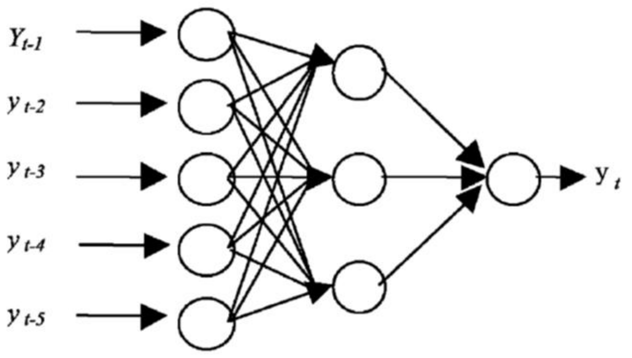

Figure 1: ANN Architecture

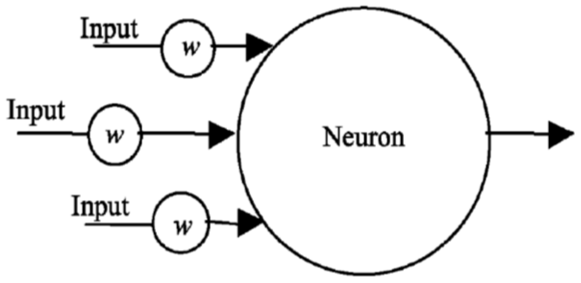

Figure 2: A Neuron

The summed values (one per neuron in the layer) are then forwarded, via a transformation function, to the next layer of the network. The output for the final layer corresponds to the network's calculation as to what the output vector should be. Initially all the weights in the network are set to random values and the network 'learns' by adjusting the weights in such a way as to reduce the difference between the network's calculation of what the output vector should be and the actual value.

After the ANN has been trained it is then tested. Testing involves presenting the network with a series of input vectors for which the tester, but not the network, knows the corresponding output values. The answers given by the network as to what it determines to be the level of output, given the presented input vector, can then be used by the tester to determine the robustness of the training process. If the robustness of the network is deemed sufficient, the network can be used in a truly predictive capacity. NN's are able to solve effectively and efficiently a wide range of problems that are either difficult or impossible to solve using 'conventional' techniques [5].

Artificial Neuron

An artificial neuron is composed of five main parts: inputs, weights, sum function, activation function and outputs. Inputs are the information that enters the cell from other cells or from the external world. Weights are values that express the effects of an input set or another process element in the previous layer on this process element. Sum function is a function that calculates the net input that comes to a cell [6]. The weighted sums of the input components (net)*j* are calculated by using equation 1:

$$(net)j = \sum_{i=1}^{n} w_{ij}x_i + b \qquad (1)$$

Where $(net)_j$ is the weight of the *j.* neuron for the input received from the preceding layer with *n.* neurons, w_{ij} is the weight between the *j.* neuron in the preceding layer, x*j* is the output of the *i.* neuron in the preceding layer [7]. *b* is a constant which is an internal addition and ∑ represents sum function. The activation function processes the net input obtained from the sum function and determines the cell output. In general for multi-layer receptive models, the sigmoid activation function (*f*(net)) is used. The output of the *j.* neuron (out)*j* is calculated employing Equation 2 with a sigmoid function as follows;

$$(\mathrm{out})\mathrm{j} = \mathrm{f}(\mathrm{net})\mathrm{j} = 1/\left(1 + \mathrm{e}^{-\propto(\mathrm{net})\mathrm{j}}\right) \qquad (2)$$

Where α is a constant which is used to control the slope of the semi-linear region. The sigmoid non-linearity activates in every layer except in the input layer [6]. The sigmoid function represented by Equation (2) gives inputs in (0, 1). As the sigmoid processor represents a continuous function, it is especially used in non-linear descriptions because its derivatives can be determined easily with regard to the parameters within the (net)*j* variable.

2.5 Developing the Neural –Based Model

The basic strategy for developing a neural-based model of material behavior is to train a neural network on the results of a series of experiments on a material (i.e. creating a database of examples). If the experimental results contain the relevant information about the material behavior, then the trained neural network would contain sufficient information about the material behavior to qualify as a material model. Such a trained neural network not only would be able to reproduce the experimental results it was trained on, but through its generalization capability it should be able to approximate the results of other experiments [8].

Most research that used ANN to model material behavior is based on back-propagation networks (BPN). Pala et al (2007) stated that the Back-propagation algorithm is one of the well-known training algorithms for the multi-layer perceptron (MLP); as it is a gradient descent technique which minimizes the error for a particular training pattern in which it adjusts the weighs by a small amount at a time. The BPN learns by comparing its output of each input pattern with a target output of that pattern, then calculating the error and propagating an error function backward through the net.

To run the network after it is trained, the values for the input parameters for the project are presented to the network. The network then calculates the node outputs by using the existing weight values and thresholds developed in the training process. To test the accuracy of the trained network, the coefficient of determination R^2 is adopted. The coefficient is a measure of how well the independent variables took into account the measured dependent variable, and denotes the strength of the linear association between *x* and *y.* It is the ratio of the explained variation to the total variation, and such that $0<R^2<1$; the higher the value of R^2, the better the prediction relation. It is useful, because it gives the proportion of the variance (fluctuation) of one variable that is predictable from the other variable.

MATERIALS AND METHODS

Concrete Materials.

The various materials that would be used in the preparation of the concrete specimens are:

-Portland Cement (Grade 32.5)

-Coarse aggregates of size 10mm and 20mm obtained from a quarry site. The aggregates were cubical in shape.

-Fine Aggregate: river sand which is free from organic materials and clay particles was used.

-Water: clean and drinkable water was used for the mix.

Mix Design:

The DOE Method of Concrete Mix Design was used to ascertain the proportions of materials in the mixes to be prepared. The method is applicable to concrete for most purposes, including roads. The method has been used for designing concrete containing fly ash (FA), or granulated blast furnace slag (GGBS).

Weight batching was used while the mixing was done by hand. Compaction of the samples was attained by tamping the moulds in a uniform specified manner.

Developing the Feed Forward Neural Network

In developing the Neural Network for compressive strength and slump prediction, the following parameters were used as input data; size of aggregate (SA), weight of cement (C), weight of water (W), weight of sand (S), and weight of crushed stone (CS). The quantities of the materials were varied to determine their resultant effect on the slump and compressive strength of the resulting concrete after curing by immersion for 28days. Minimum and maximum values of the constituent materials were determined from the mix design.

The input layer consisted of 5 (five) inputs while the output layer consisted of two outputs; namely, compressive strength and slump. The other parameters needed in the model development included; number of hidden layers, number of neurons in hidden layers and learning rate were determined as the model was simulated.

Sample Size

A total of 216 (100×100×100mm size) cubes of concrete were casted and used in the training, testing and validation of the model. 60% of the data was randomly selected and used for the training while 20% each were used for the testing and validation of the model.

RESULTS AND DISCUSSION

Concrete Slump and Compressive Strengths:

The concrete produced had varying water/cement ratios thereby leading to varying slumps and compressive strengths. The slumps varied from 0 to 130 while the compressive strength of the samples was between 11- 30N/mm^2.

Training, Testing and Validation of Specimen Data:

The network was successfully trained. The Mean Square Error (MSE) obtained after the training of the network was 0.0134525. Other parameters used were a learning rate of 0.0001, performance goal of 0, number of training epochs was set at 10,000, while the epoch shows was set to 10. The transfer function used between the first (i.e. input layer), second and third layers was the ‘tansigmoid’ function while the ‘purelin’ transfer function was used between the third and fourth layer (i.e. output layer). The input and output vector data were divided as follows; 60 per cent was used to train the network, 20 per cent was used to validate how well the network generalized, while the last 20 per cent of the vectors provided an independent test of the network generalization to data that the network has never seen. The Scaled Conjugate Gradient algorithm (‘trainscg’) was used to train the feed forward network. In order to make the training process more efficient, the input and output data were preprocessed and scaled. The data were normalized by scaling the input and targets vectors using the mean and standard deviations of the training set. The function ***mapstd*** was used to normalize and scale the inputs and targets so that they had zero mean and unity standard deviation. The normalized inputs and targets were stored in matrices ***pn*** and ***tn*** which returned values of zero means and unity standard deviation. The settings structure ***ps*** and ***ts*** contained the means and standard deviations of the original inputs and original targets after which a Principal Component Analysis (PCA) was conducted on the input and output vectors.

With ***mapstd*** being used to scale the targets, the output of the network was trained to produce outputs with zero mean and unity standard deviation. Inorder to convert these outputs back into the same units that were used for the original targets, ***ts*** was used. The following code was used to simulate the network that was trained, and used to convert the network output back to its original units.

an = sim(net,pn);

a = mapstd('reverse',an,ts);

The network output ***an*** corresponds to the normalized targets ***tn*** while the unnormalized network output ***a*** had the same units as the original targets ***t***. Since ***mapstd*** was used to preprocess the training data set, whenever the trained network is to be used with new inputs, the data should be preprocessed with the means and standard deviations that were computed for the training set using ***ps***.

CONCLUSION

The code used for the NN are shown in Appendix A while Appendix B is a table which shows the network inputs, targets and predicted outcomes from the model.

The following code can be employed when new set of inputs are to be introduced to the already trained network.

pnewn = mapstd(`apply',pnew,ps);

anewn = sim(net,pnewn);

anew = mapstd(`reverse',anewn,ts);

The following graphs and plots were also obtained after the neural network was trained.

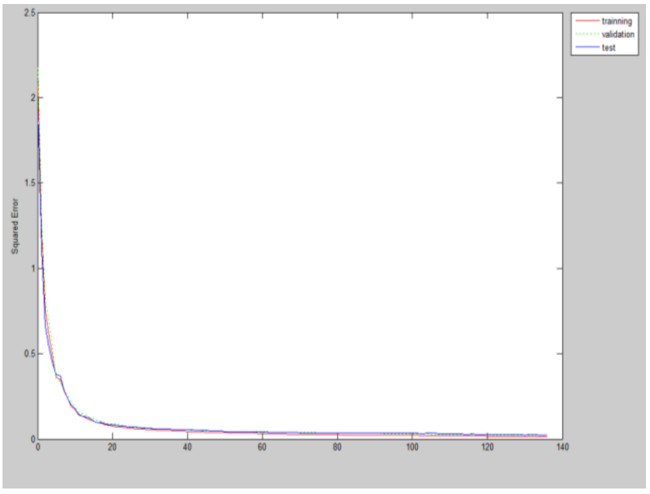

Figure 3: Mean Squared Error against Training Epochs

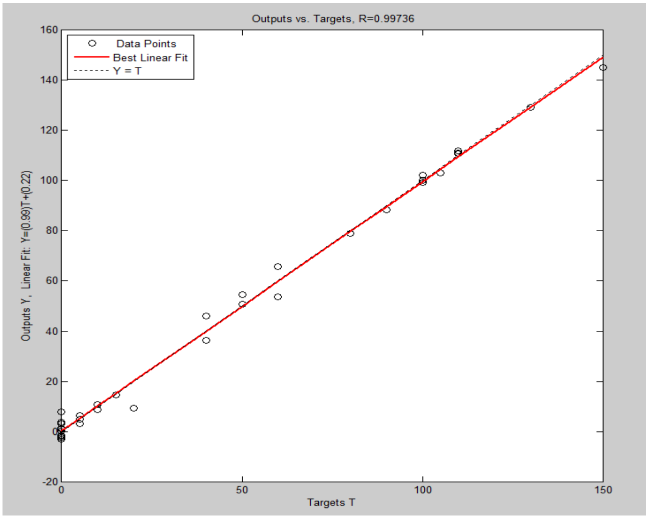

Figure 4: Slump: Validation Set against Target Outputs of Concrete Slumps.

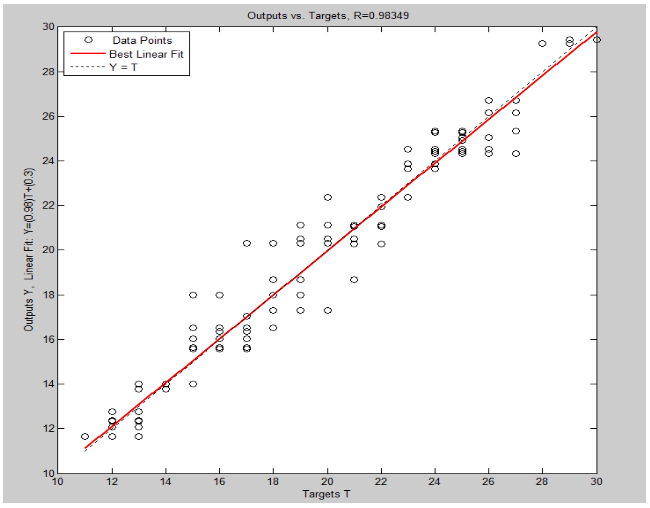

Figure 5: Training Set against Target Outputs of Compressive Strengths

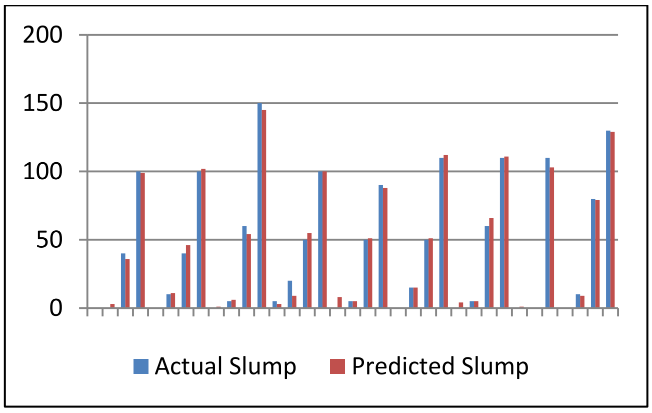

Figure 6: Comparison of Actual and Predicted Slump (mm)

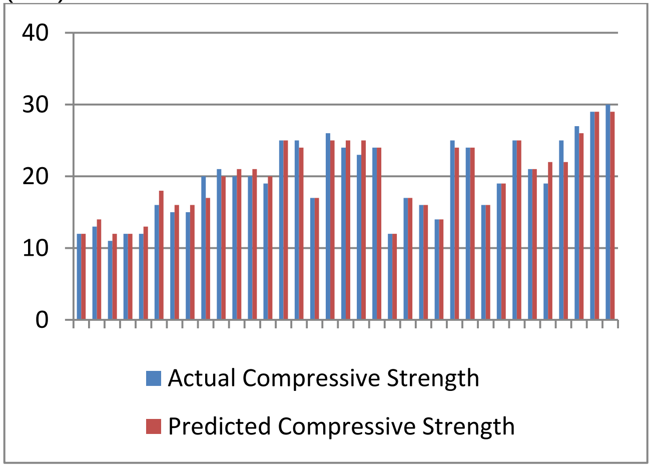

Figure 7: Comparison of Actual and Predicted Compressive Strength (N/mm^2).

REFERENCES

[1]-Oloyede, S.A; C.B. Omoogun, O.A. Akinjare (2010): "Tackling Causes of Frequent Building Collapse in Nigeria" *Journal of Sustainable Development: Canadian Center of Science and Education*; Vol.3, No.3, September 2010, pp 127-132.

[2]-Neville, A. M. (1981). Properties of Concrete. Longman Scientific and Technical, London, 717p.

[3]-Tazelaar, J. M. (1989). Neural Networks. *Journal BYTE*, 14 (8). 214.

[4]-Zurada, J. M. (1992). Introduction to Artificial Neural Network Systems. West Publishing Company, Boston Massachusetts, p. 58

[5]-Bishop, C.M., (1992). Neural Networks for Pattern Recognition. Oxford University Press, London. p. 83

[6]-Kewalranmi, M. A., and Grupta, R. (2006). Concrete Compressive Strength Prediction using Pulse Velocity through Artificial Neural Networks, *Automation in Construction.* 15 (3), 374-379.

[7]-Pala M., Ozbay E., Oztas A., Ishak Yuce M., Construction and Building Materials 21 (2) (2007) 384-394

[8]-Jung, H. C., and Jamshid, G., (2001). Genetic Algorithm in Structural Damage Detection. *Computers and Structures*. 79 (30), 1335- 53.

[9]-BS 1881: Part 1: 1970. Methods of Sampling Fresh Concrete. British Standards Institute.

[10]-BS 1881: Part 2: 1970. Methods of Testing Fresh Concrete. British Standard Institute.

[11]-BS 1881: Part 3: 1970. Methods of making and Curing Concrete Test Specimens. British Standard Institute.

[12]-BS 1881: Part 4: 1970. Methods of Testing Concrete for Strength. British Standard Institute.

[13]-BS 882 & 1201: 1965. Specification for Aggregates from Natural Sources for Concrete (including Granolithic). British Standard Institute.

[14]-EN 197-1: 2000, Cement : Part 1. Composition, Specifications and Conformity Criteria for Common Cements.

[15]-Mosley, W. H., Bungey, J. H., and Hulse, R. (1999), Reinforced Concrete Design 5th Ed. Palgrave Publishers Ltd, New York. Pp. 11-13

[16]-NIS 444: Part 1: 2003. Composition, Specification and Conformity Criteria for Common Cements. Standards Organisation of Nigeria.

[17]-NIS 446: 2003. Methods of Testing Cement; Determination of Strength. Standards Organisation of Nigeria.

[18]-Shetty, M.S. (2005), Concrete Technology, Theory and Practice. S. Chand & Company Ltd, India. 503p.

[19]-Tazawa, E. and Miyazawa, S. (1996). Influence of Cement and Admixture on Autegenous Shrinkage of Cement Paste. *Cement and Concrete Composites,* 25 (2), 281 – 287.

APPENDIX

A. MATLAB Code used in Training Neural Network.

```
A = [10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
10 10 10 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20];
```

```
>> B = [180 180 180 180 180 180 205 205 205
205 205 205 230 230 230 230 230 230 250 250 250
250 250 250 180 180 180 180 180 180 205 205 205
205 205 205 230 230 230 230 230 230 250 250 250
250 250 250 180 180 180 180 180 180 205 205 205
205 205 205 230 230 230 230 230 230 250 250 250
250 250 250 180 180 180 180 180 180 205 205 205
205 205 205 230 230 230 230 230 230 250 250 250
250 250 250 180 180 180 180 180 180 205 205 205
205 205 205 230 230 230 230 230 230 250 250 250
250 250 250 170 170 170 170 170 170 190 190 190
190 190 190 210 210 210 210 210 210 225 225 225
225 225 225 170 170 170 170 170 170 190 190 190
190 190 190 210 210 210 210 210 210 225 225 225
225 225 225 170 170 170 170 170 170 190 190 190
190 190 190 210 210 210 210 210 210 225 225 225
225 225 225 170 170 170 170 170 170 190 190 190
190 190 190 210 210 210 210 210 210 225 225 225
225 225 225];
```

```
>> C = [225 225 225 225 225 225 256 256 256
256 256 256 288 288 288 288 288 288 313 313 313
313 313 313 247 247 247 247 247 247 281 281 281
281 281 281 315 315 315 315 315 315 342 342 342
342 342 342 277 277 277 277 277 277 315 315 315
315 315 315 354 354 354 354 354 354 385 385 385
385 385 385 300 300 300 300 300 300 342 342 342
342 342 342 383 383 383 383 383 383 417 417 417
417 417 417 327 327 327 327 327 327 373 373 373
373 373 373 418 418 418 418 418 418 455 455 455
455 455 455 180 180 180 180 180 180 238 238 238
238 238 238 263 263 263 263 263 263 281 281 281
281 281 281 275 275 275 275 275 275 292 292 292
292 292 292 323 323 323 323 323 323 346 346 346
346 346 346 300 300 300 300 300 300 317 317 317
317 317 317 350 350 350 350 350 350 375 375 375
375 375 375 325 325 325 325 325 325 352 352 352
352 352 352 389 389 389 389 389 389 417 417 417
417 417 417];
```

```
>> D = [818 818 818 818 818 818 853 853 853
853 853 853 889 889 889 889 889 889 894 894 894
894 894 894 809 809 809 809 809 809 785 785 785
785 785 785 858 858 858 858 858 858 844 844 844
844 844 844 777 777 777 777 777 777 771 771 771
771 771 771 804 804 804 804 804 804 815 815 815
815 815 815 749 749 749 749 749 749 723 723 723
723 723 723 773 773 773 773 773 773 808 808 808
808 808 808 699 699 699 699 699 699 711 711 711
711 711 711 775 775 775 775 775 775 806 806 806
806 806 806 742 742 742 742 742 742 708 708 708
708 708 708 732 732 732 732 732 732 825 825 825
825 825 825 608 608 608 608 608 608 610 610 610
610 610 610 654 654 654 654 654 654 706 706 706
706 706 706 600 600 600 600 600 600 602 602 602
602 602 602 644 644 644 644 644 644 694 694 694
694 694 694 553 553 553 553 553 553 572 572 572
572 572 572 594 594 594 594 594 594 678 678 678
678 678 678];
```

```
>> E = [1227 1227 1227 1227 1227 1227 1086
1086 1086 1086 1086 1086 963 963 963 963 963 963
894 894 894 894 894 894 1214 1214 1214 1214 1214
1214 1129 1129 1129 1129 1129 1129 967 967 967
967 967 967 914 914 914 914 914 914 1216 1216
1216 1216 1216 1216 1109 1109 1109 1109 1109
1109 982 982 982 982 982 982 900 900 900 900 900
900 1221 1221 1221 1221 1221 1221 1130 1130
1130 1130 1130 1130 984 984 984 984 984 984 875
875 875 875 875 875 1244 1244 1244 1244 1244
1244 1111 1111 1111 1111 1111 1111 947 947 947
947 947 947 839 839 839 839 839 839 1378 1378
1378 1378 1378 1378 1314 1314 1314 1314 1314
1314 1195 1195 1195 1195 1195 1195 1049 1049
1049 1049 1049 1049 1417 1417 1417 1417 1417
1417 1358 1358 1358 1358 1358 1358 1213 1213
1213 1213 1213 1213 1103 1103 1103 1103 1103
1103 1400 1400 1400 1400 1400 1400 1341 1341
1341 1341 1341 1341 1196 1196 1196 1196 1196
1196 1086 1086 1086 1086 1086 1086 1422 1422
1422 1422 1422 1422 1336 1336 1336 1336 1336
1336 1207 1207 1207 1207 1207 1207 1060 1060
1060 1060 1060 1060];
```

```
>> p = vertcat(A,B,C,D,E);
```

```
>> F = [0 0 0 0 0 0 0 0 0 0 0 0 40 40 40 40 40 40
100 100 100 100 100 100 0 0 0 0 0 0 10 10 10 10 10
10 40 40 40 40 40 40 100 100 100 100 100 100 0 0 0
0 0 0 5 5 5 5 5 60 60 60 60 60 60 150 150 150 150
150 150 5 5 5 5 5 5 20 20 20 20 20 20 50 50 50 50 50
50 100 100 100 100 100 100 0 0 0 0 0 0 5 5 5 5 5 5
50 50 50 50 50 50 90 90 90 90 90 90 0 0 0 0 0 0 15
15 15 15 15 15 50 50 50 50 50 50 110 110 110 110
110 110 0 0 0 0 0 0 5 5 5 5 5 5 60 60 60 60 60 110
110 110 110 110 110 0 0 0 0 0 0 0 0 0 0 0 0 105 105
105 105 105 105 110 110 110 110 110 110 0 0 0 0 0
0 10 10 10 10 10 10 80 80 80 80 80 80 130 130 130
130 130 130];
```

```
>> G = [12 12 13 13 12 13 13 14 14 14 13 13 11
13 12 12 11 12 12 13 12 13 13 13 12 12 12 13 13 12
16 15 19 19 18 18 15 15 17 17 16 17 15 16 16 15 17
15 20 20 18 19 18 20 21 21 22 22 22 22 20 21 22 22
20 19 20 19 21 19 20 21 19 17 18 18 19 20 25 24 25
24 25 25 25 25 24 27 25 26 17 16 17 17 15 18 26 26
25 25 26 26 24 25 27 25 24 25 23 23 24 24 26 25 24
24 23 23 24 24 12 13 12 13 12 13 17 17 17 17 17 17
16 16 16 16 17 16 14 14 14 15 14 13 25 24 25 24 24
24 24 24 24 24 23 24 16 15 16 17 17 17 19 18 18 19
19 21 25 25 25 25 25 25 21 21 22 21 21 22 22 22 23
20 22 22 22 22 22 22 22 22 27 27 26 26 27 26 27 26
26 26 26 26 29 29 29 29 29 28 30 30 29 30 29 30];
```

```
>> t = vertcat(F,G);
>> [pn,pp1] = mapstd (p);
>> [tn,tp] = mapstd (t);
>> [ptrans,pp2] = processpca (pn,0.001);
>> [R,Q] = size (ptrans);
>> iitst = 2:4:Q;
>> iival = 4:4:Q;
>> iitr = [1:4:Q 3:4:Q];
>> validation.P = ptrans (:,iival);
>> validation.T = tn (:,iival);
>> testing.P = ptrans (:,iitst);
>> testing.T = tn (:,iitst);
>> ptr = ptrans (:,iitr);
>> ttr = tn (:,iitr);
>> net = newff(minmax(ptr),[10 8 2],{'tansig' 'tansig'
'purelin'}, 'trainscg');
>> net.trainParam.show = 10;
>> net.trainParam.lr = 0.0001;
>> net.trainParam.epochs = 10000;
>> net.trainParam.goal = 0;
>> [net,tr] = train(net,ptr,ttr,[],[],validation,testing);
```

A = Coarse Aggregate Size, B = Water Content (g), C = Cement Content (g), D = Fine Aggregate Weight (g), E = Coarse Aggregate Weight (g), F = Experimental Slump Values (mm), G = Experimental Compressive Strength (N/mm^2), H = Predicted Slump Values (mm), J = Predicted Compressive Strength Values (N/mm^2).

B. Network Inputs, Targets And Predicted Outcomes From The Model

S/N	A	B	C	D	E	F	G	H	J
1	10	180	225	818	1227	0	12	-2	12
2	10	180	225	818	1227	0	12	-2	12
3	10	180	225	818	1227	0	13	-2	12
4	10	180	225	818	1227	0	13	-2	12
5	10	180	225	818	1227	0	12	-2	12
6	10	180	225	818	1227	0	13	-2	12
7	10	205	256	853	1086	0	13	3	14
8	10	205	256	853	1086	0	14	3	14
9	10	205	256	853	1086	0	14	3	14
10	10	205	256	853	1086	0	14	3	14
11	10	205	256	853	1086	0	13	3	14
12	10	205	256	853	1086	0	13	3	14
13	10	230	288	889	963	40	11	36	12
14	10	230	288	889	963	40	13	36	12
15	10	230	288	889	963	40	12	36	12
16	10	230	288	889	963	40	12	36	12
17	10	230	288	889	963	40	11	36	12
18	10	230	288	889	963	40	12	36	12
19	10	250	313	894	894	100	12	99	12
20	10	250	313	894	894	100	13	99	12
21	10	250	313	894	894	100	12	99	12
22	10	250	313	894	894	100	13	99	12
23	10	250	313	894	894	100	13	99	12
24	10	250	313	894	894	100	13	99	12
25	10	180	247	809	1214	0	12	-3	13
26	10	180	247	809	1214	0	12	-3	13
27	10	180	247	809	1214	0	12	-3	13
28	10	180	247	809	1214	0	13	-3	13
29	10	180	247	809	1214	0	13	-3	13
30	10	180	247	809	1214	0	12	-3	13
31	10	205	281	785	1129	10	16	11	18
32	10	205	281	785	1129	10	15	11	18
33	10	205	281	785	1129	10	19	11	18
34	10	205	281	785	1129	10	19	11	18
35	10	205	281	785	1129	10	18	11	18
36	10	205	281	785	1129	10	18	11	18
37	10	230	315	858	967	40	15	46	16
38	10	230	315	858	967	40	15	46	16
39	10	230	315	858	967	40	17	46	16
40	10	230	315	858	967	40	17	46	16
41	10	230	315	858	967	40	16	46	16
42	10	230	315	858	967	40	17	46	16
43	10	250	342	844	914	100	15	102	16
44	10	250	342	844	914	100	16	102	16
45	10	250	342	844	914	100	16	102	16
46	10	250	342	844	914	100	15	102	16
47	10	250	342	844	914	100	17	102	16
48	10	250	342	844	914	100	15	102	16
49	10	180	277	777	1216	0	20	1	17
50	10	180	277	777	1216	0	20	1	17
51	10	180	277	777	1216	0	18	1	17
52	10	180	277	777	1216	0	19	1	17
53	10	180	277	777	1216	0	18	1	17
54	10	180	277	777	1216	0	20	1	17
55	10	205	315	771	1109	5	21	6	20
56	10	205	315	771	1109	5	21	6	20
57	10	205	315	771	1109	5	22	6	20
58	10	205	315	771	1109	5	22	6	20
59	10	205	315	771	1109	5	22	6	20
60	10	205	315	771	1109	5	22	6	20
61	10	230	354	804	982	60	20	54	21
62	10	230	354	804	982	60	21	54	21
63	10	230	354	804	982	60	22	54	21
64	10	230	354	804	982	60	22	54	21
65	10	230	354	804	982	60	20	54	21
66	10	230	354	804	982	60	19	54	21
67	10	250	385	815	900	150	20	145	21
68	10	250	385	815	900	150	19	145	21
69	10	250	385	815	900	150	21	145	21
70	10	250	385	815	900	150	19	145	21
71	10	250	385	815	900	150	20	145	21
72	10	250	385	815	900	150	21	145	21
73	10	180	300	749	1221	5	19	3	20
74	10	180	300	749	1221	5	17	3	20
75	10	180	300	749	1221	5	18	3	20
76	10	180	300	749	1221	5	18	3	20
77	10	180	300	749	1221	5	19	3	20
78	10	180	300	749	1221	5	20	3	20
79	10	205	342	723	1130	20	25	9	25
80	10	205	342	723	1130	20	24	9	25
81	10	205	342	723	1130	20	25	9	25
82	10	205	342	723	1130	20	24	9	25
83	10	205	342	723	1130	20	25	9	25
84	10	205	342	723	1130	20	25	9	25
85	10	230	383	773	984	50	25	55	24

86	10	230	383	773	984	50	25	55	24
87	10	230	383	773	984	50	24	55	24
88	10	230	383	773	984	50	27	55	24
89	10	230	383	773	984	50	25	55	24
90	10	230	383	773	984	50	26	55	24
91	10	250	417	808	875	100	17	100	17
92	10	250	417	808	875	100	16	100	17
93	10	250	417	808	875	100	17	100	17
94	10	250	417	808	875	100	17	100	17
95	10	250	417	808	875	100	15	100	17
96	10	250	417	808	875	100	18	100	17
97	10	180	327	699	1244	0	26	8	25
98	10	180	327	699	1244	0	26	8	25
99	10	180	327	699	1244	0	25	8	25
100	10	180	327	699	1244	0	25	8	25
101	10	180	327	699	1244	0	26	8	25
102	10	180	327	699	1244	0	26	8	25
103	10	205	373	711	1111	5	24	5	25
104	10	205	373	711	1111	5	25	5	25
105	10	205	373	711	1111	5	27	5	25
106	10	205	373	711	1111	5	25	5	25
107	10	205	373	711	1111	5	24	5	25
108	10	205	373	711	1111	5	25	5	25
109	10	230	418	775	947	50	23	51	25
110	10	230	418	775	947	50	23	51	25
111	10	230	418	775	947	50	24	51	25
112	10	230	418	775	947	50	24	51	25
113	10	230	418	775	947	50	26	51	25
114	10	230	418	775	947	50	25	51	25
115	10	250	455	806	839	90	24	88	24
116	10	250	455	806	839	90	24	88	24
117	10	250	455	806	839	90	23	88	24
118	10	250	455	806	839	90	23	88	24
119	10	250	455	806	839	90	24	88	24
120	10	250	455	806	839	90	24	88	24
121	20	170	180	742	1378	0	12	-2	12
122	20	170	180	742	1378	0	13	-2	12
123	20	170	180	742	1378	0	12	-2	12
124	20	170	180	742	1378	0	13	-2	12
125	20	170	180	742	1378	0	12	-2	12
126	20	170	180	742	1378	0	13	-2	12
127	20	190	238	708	1314	15	17	15	17
128	20	190	238	708	1314	15	17	15	17
129	20	190	238	708	1314	15	17	15	17
130	20	190	238	708	1314	15	17	15	17
131	20	190	238	708	1314	15	17	15	17
132	20	190	238	708	1314	15	17	15	17
133	20	210	263	732	1195	50	16	51	16
134	20	210	263	732	1195	50	16	51	16
135	20	210	263	732	1195	50	16	51	16
136	20	210	263	732	1195	50	16	51	16
137	20	210	263	732	1195	50	17	51	16
138	20	210	263	732	1195	50	16	51	16
139	20	225	281	825	1049	110	14	112	14
140	20	225	281	825	1049	110	14	112	14
141	20	225	281	825	1049	110	14	112	14
142	20	225	281	825	1049	110	15	112	14
143	20	225	281	825	1049	110	14	112	14
144	20	225	281	825	1049	110	13	112	14
145	20	170	275	608	1417	0	25	4	24
146	20	170	275	608	1417	0	24	4	24
147	20	170	275	608	1417	0	25	4	24
148	20	170	275	608	1417	0	24	4	24
149	20	170	275	608	1417	0	24	4	24
150	20	170	275	608	1417	0	24	4	24
151	20	190	292	610	1358	5	24	5	24
152	20	190	292	610	1358	5	24	5	24
153	20	190	292	610	1358	5	24	5	24
154	20	190	292	610	1358	5	24	5	24
155	20	190	292	610	1358	5	23	5	24
156	20	190	292	610	1358	5	24	5	24
157	20	210	323	654	1213	60	16	66	16
158	20	210	323	654	1213	60	15	66	16
159	20	210	323	654	1213	60	16	66	16
160	20	210	323	654	1213	60	17	66	16
161	20	210	323	654	1213	60	17	66	16
162	20	210	323	654	1213	60	17	66	16
163	20	225	346	706	1103	110	19	111	19
164	20	225	346	706	1103	110	18	111	19
165	20	225	346	706	1103	110	18	111	19
166	20	225	346	706	1103	110	19	111	19
167	20	225	346	706	1103	110	19	111	19
168	20	225	346	706	1103	110	21	111	19
169	20	170	300	600	1400	0	25	1	25
170	20	170	300	600	1400	0	25	1	25
171	20	170	300	600	1400	0	25	1	25
172	20	170	300	600	1400	0	25	1	25
173	20	170	300	600	1400	0	25	1	25
174	20	170	300	600	1400	0	25	1	25
175	20	190	317	602	1341	0	21	0	21
176	20	190	317	602	1341	0	21	0	21
177	20	190	317	602	1341	0	22	0	21
178	20	190	317	602	1341	0	21	0	21
179	20	190	317	602	1341	0	21	0	21
180	20	190	317	602	1341	0	22	0	21
181	20	225	346	706	1103	110	19	103	22

182	20	225	346	706	1103	110	18	103	22
183	20	225	346	706	1103	110	18	103	22
184	20	225	346	706	1103	110	19	103	22
185	20	225	346	706	1103	110	19	103	22
186	20	225	346	706	1103	110	21	103	22
187	20	170	300	600	1400	0	25	111	22
188	20	170	300	600	1400	0	25	111	22
189	20	170	300	600	1400	0	25	111	22
190	20	170	300	600	1400	0	25	111	22
191	20	170	300	600	1400	0	25	111	22
192	20	170	300	600	1400	0	25	111	22
193	20	190	317	602	1341	0	21	-2	27
194	20	190	317	602	1341	0	21	-2	27
195	20	190	317	602	1341	0	22	-2	27
196	20	190	317	602	1341	0	21	-2	27
197	20	190	317	602	1341	0	21	-2	27
198	20	190	317	602	1341	0	22	-2	27
199	20	190	352	572	1336	10	27	9	26
200	20	190	352	572	1336	10	26	9	26
201	20	190	352	572	1336	10	26	9	26
202	20	190	352	572	1336	10	26	9	26
203	20	190	352	572	1336	10	26	9	26
204	20	190	352	572	1336	10	26	9	26
205	20	210	389	594	1207	80	29	79	29
206	20	210	389	594	1207	80	29	79	29
207	20	210	389	594	1207	80	29	79	29
208	20	210	389	594	1207	80	29	79	29
209	20	210	389	594	1207	80	29	79	29
210	20	210	389	594	1207	80	28	79	29
211	20	225	417	678	1060	130	30	129	29
212	20	225	417	678	1060	130	30	129	29
213	20	225	417	678	1060	130	29	129	29
214	20	225	417	678	1060	130	30	129	29
215	20	225	417	678	1060	130	29	129	29
216	20	225	417	678	1060	130	30	129	29