# Artificial Neural Networks to Determine Source of Acoustic Emission and Damage Detection

**Mehrdad Shafiei Dizaji[1], Farzad Shafiei Dizaji[2]**

[1] *Former graduate, Department of Civil Engineering, Sharif University of Technology, Tehran, Iran*
[2] *Former graduate, Department of Civil Engineering, Iran University of Science & Technology, Tehran, Iran*

---

**Abstract -** *This paper explains how multi-layer perceptrons have been used in the post processing of data collected by 16 sensors during a number of controlled sonic tests as well as surrounding noise effect on a concrete wall. The data has been divided into 3 main types: Type I, result of controlled tests related to energy release from cracking inside the concrete wall as a result of applied loading on the top of the wall. Type II, Waveform data created by transmitting a signal from one of the pulsing sensors to all receiver sensors. Type III, Waveform data not related to induced microcracking inside the concrete, such as human/mechanical activity on or near the concrete beam. The data available for this preliminary study contained 4 tests of type I, 1 test of type II and 8 tests of type III. The results show that the designed neural network is successful in the identification of the different types of waveform data.*

***Key Words***: neural networks, microcracking, concrete wall, wave propagation.

## 1. METHODOLOGY OF USING NEURAL NETWORKS FOR IDENTIFICATION OF RECORDS

Multi-layer Feed Forward Neural Networks (MLFFNNs) have been utilized in the post processing of data collected by 16 sensors during a number of controlled sonic tests as well as surrounding noise effect on a concrete wall.

The data consists of three main types: Type I, result of controlled tests related to energy release from cracking inside the concrete wall as a result of applied loading on the top of the wall. Type II, Waveform data created by transmitting a signal from one of the pulsing sensors to all receiver sensors. Type III, Waveform data not related to induced microcracking inside the concrete, such as human/mechanical activity on or near the concrete beam. The data available for this preliminary study contained 4 tests of type I, 1 test of type II and 8 tests of type III.

*Assumption:* The main idea and assumption has been that there is a strong and specific correlation between the records of different sensors for type I events, while this correlation cannot be seen for the other two types of events. So if a neural network is trained to learn the correlation between the record of a sensor with the record of another sensor based on data collected from type I tests, this neural network can be used to determine if a new record is of type I or else.

The proposed method based on artificial neural networks is capable of distinguishing the records corresponding to energy release due to induced cracks within the concrete wall from the records corresponding to environmental noise and from other sources. However, it is early to consider this technique as a proven technique for identification of data types. More work is needed.

### 1.1 ARTIFICIAL NEURAL NETWORKS

Multilayer neural networks which are sometimes referred to as perceptrons, are simple models of several connected neurons similar to what is seen in the natural neural networks of animals. The main objective of building these artificial models of brain has been to design systems which can show some learning capabilities like the natural brain [1]. A simple model of perceptrons can be seen in Figure 1.

The network is generally comprised of an input, an output and one or more inside layers of neurons. The neurons are connected in a feed-forward manner, i.e. the neurons in each layer are connected to the neurons in its immediate previous and next layers. Mainly the connections of a neural network are the adaptive adjustable parts of it. Each neuron is a processing unit. Given an input vector to its input layer, the input signals propagate forward inside the neural network until it reaches the output layer. The vector of signals which appears in the output layer is considered as the output vector of the neural network, associated with the given input. The training of a neural network is the procedure of gradual modification of the connection weights until the output from a given input vector is close to the desirable target output vector.

When it is desired to build a neural network to learn an ordered set of many input-output vectors, the training and learning procedure can be very complicated and might even diverge or converge to a set of connection weights for which the neural network cannot produce desirable predictions, in which case the modelling fails. The training of the neural networks on data representing nonlinear behaviour of materials and systems, like the problem of this paper, is generally challenging. Multilayer Feed-Forward Neural Networks (MFFNNs) as general approximator tools have been found a special place in the engineering applications [1-10].

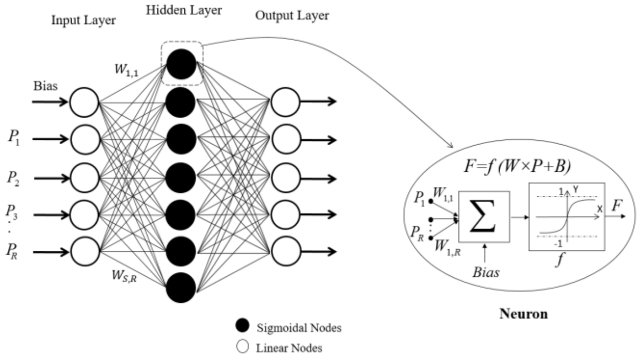


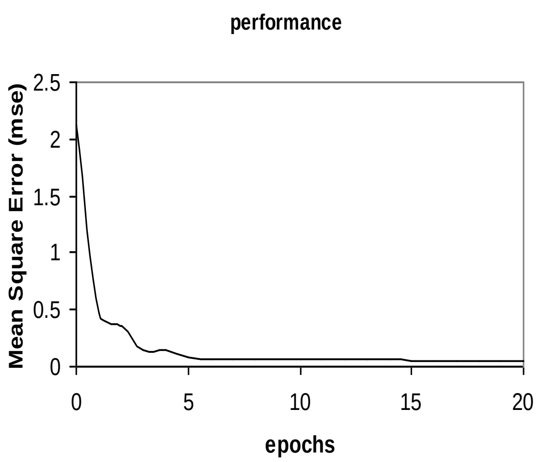

Figure 1. An architecture of a neural network

## 1.2 TRAINING OF A NEURAL NETWORK TO LEARN CORRELATION OF RECORDS BY SENSORS

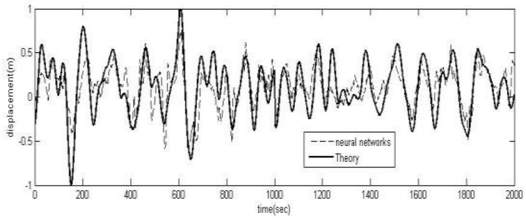
Selecting two sensors- for the time being this selection has been done arbitrarily, but it might be possible to find a rule for the selection of the sensors- a neural network is trained based on the time history of their records. Let's call these sensors 5 and 14. The time history of measurements related to these two sensors are used in the training. The record of 5 is used as input and the record of 14 as output. At each sampling time, the record of 5 is given to the neural network and it is trained to learn the record of 14 as output.

In this study, the record corresponding to each event comprised of 4096 samples at 0.004 seconds. Also there has been 4 records of type I available.2 record of these records have been used in the training. The training procedure has been done following ordinary back-propagation algorithm and convergence has been achieved easily.

For example, for A=sensor 5 and B=sensor 14, the information about training including the error and figure of training data has been placed in Figure 2. Also architecture of the trained neural network can be seen in Figure 3.

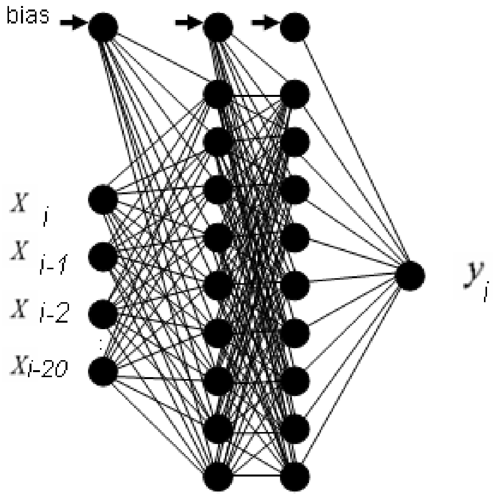



(a)

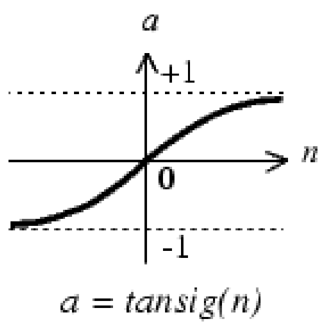



(b)

Figure 2. (a) Training error, (b) The results with training data




(a)




(b)

Figure 3. (a) Architecture of designed neural network (b) Sigmoid Transfer function

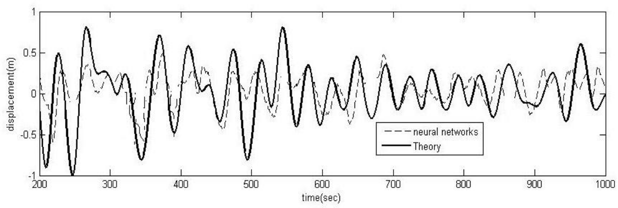



Figure 4. A=5 with B=14. Train with type1 and test with another type1

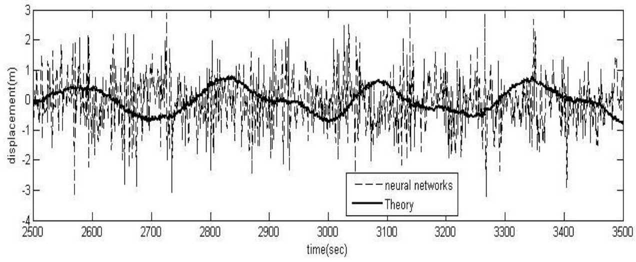



Figure 5. A=5 with B=14. Train with type1 and test with type2

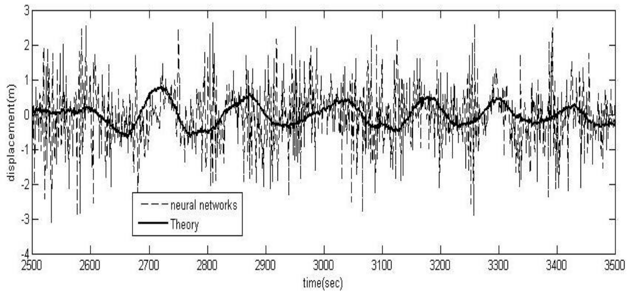



Figure 6. A=5 with B=14. Train with type1 and test with type3

## 2. TESTING TRAINED NEURAL NETWORK

The record corresponding to a test which has not been used in training of the neural network was used in its testing. To this end, the record of sensor A was given to the trained neural network and its corresponding output prediction was recorded and compared with the record from test, corresponding to sensor B. For example, the prediction and measures data for A=5 and B=14 can be seen in Figure 4. As can be seen the prediction is very similar to the real measurement of sensor B=14.

Also the same neural network was tested on the available data types II and III. It has been expected that given the records of A, the predictions by the neural network should be obviously different than the recording corresponding to B. This expectation has been satisfied. For example, again for A=5 and B=14, Figures. (5) and (6) show the result for type II and II records respectively. As can be seen the predictions are significantly different that the recorded data, leading to the conclusion that the records are not of type I.

Moreover, the proposed neural network is tested on the available data form other sensors. As can be seen from Figure 7 to 30, the model has capability of distinguishing the type of data from sensors.

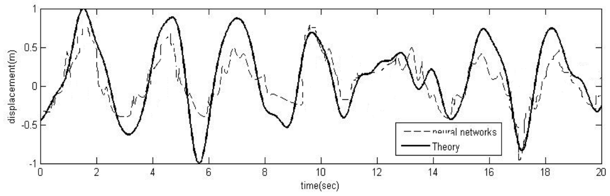



Figure 7. A=2 with B=6. Train with type1 and test with another type1

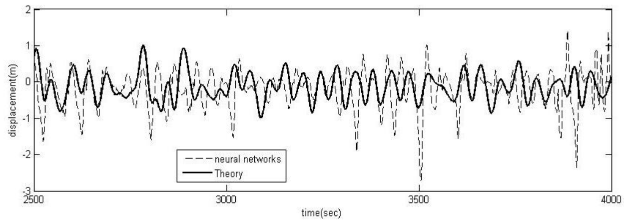



Figure 8. A=2 with B=6. Train with type1 and test with type2

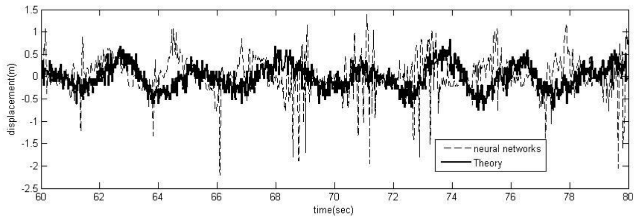



Figure 9. A=2 with B=6. Train with type1 and test with type3

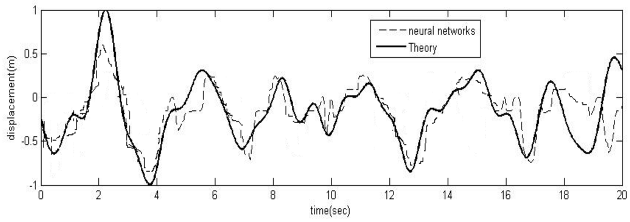



Figure 10. A=7 with B=15. Train with type1 and test with another type1

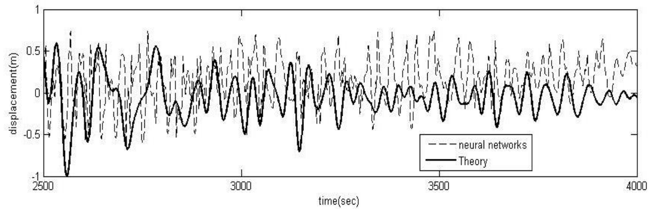



Figure 11. A=7 with B=15. Train with type1 and test with type2

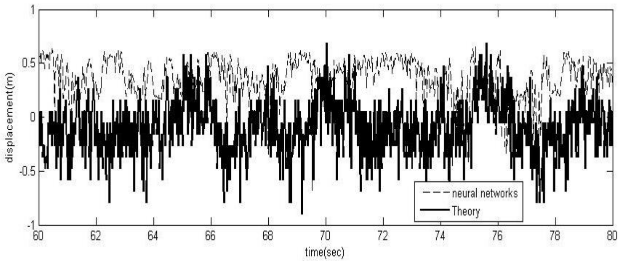



Figure 12. A=7 with B=15. Train with type1 and test with type3

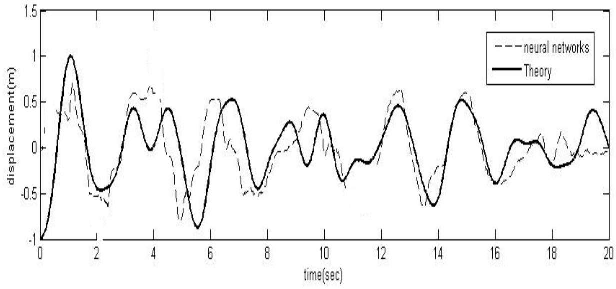



Figure 13. A=2 with B=16. Train with type1 and test with another type1

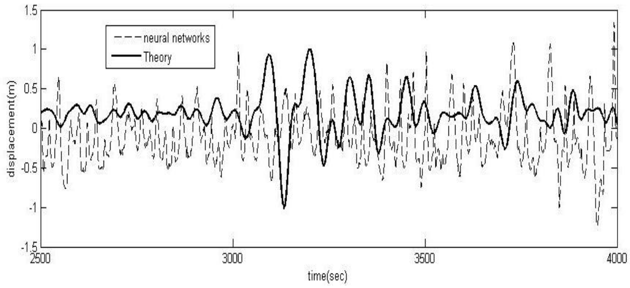



Figure 14. A=2 with B=16. Train with type1 and test with type2

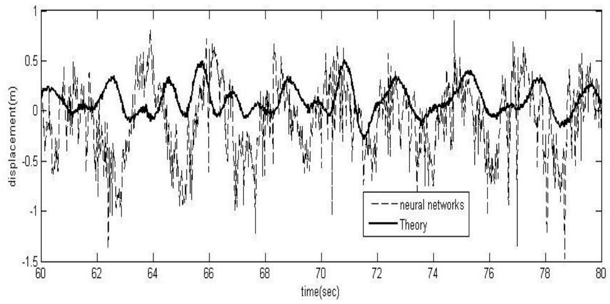



Figure 15. A=2 with B=16. Train with type1 and test with type3

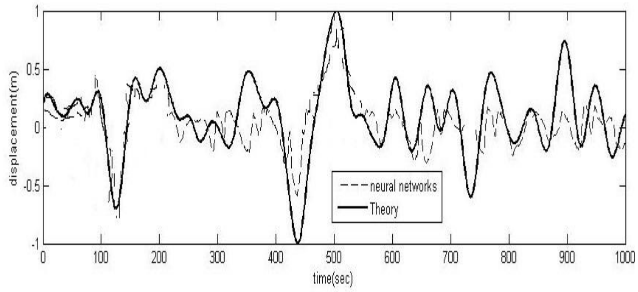



Figure 16. A=3 with B=9. Train with type1 and test with another type1

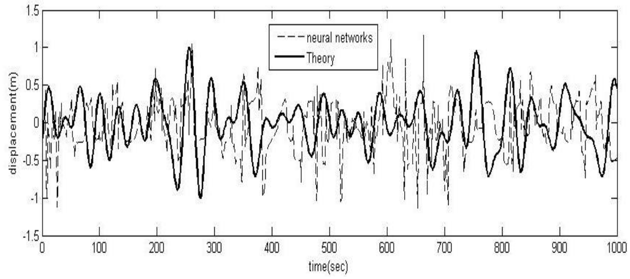



Figure 17. A=3 with B=9. Train with type1 and test with type2

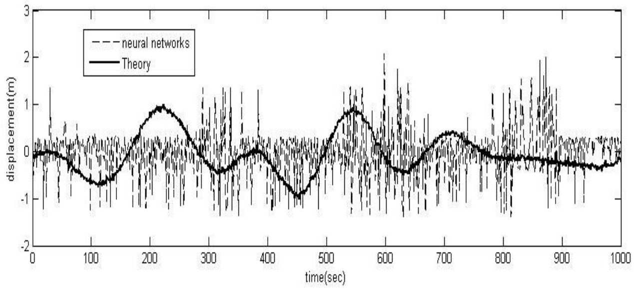



Figure 18. A=3 with B=9. Train with type1 and test with type3

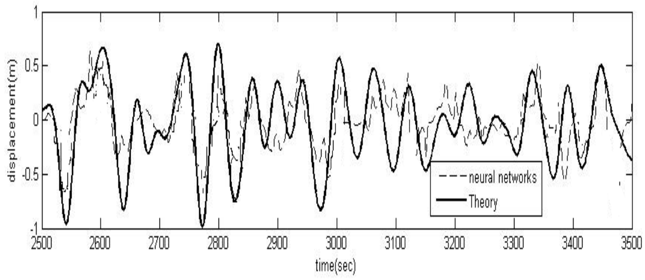



Figure 19. A=2 with B13. Train with type1 and test with another type1

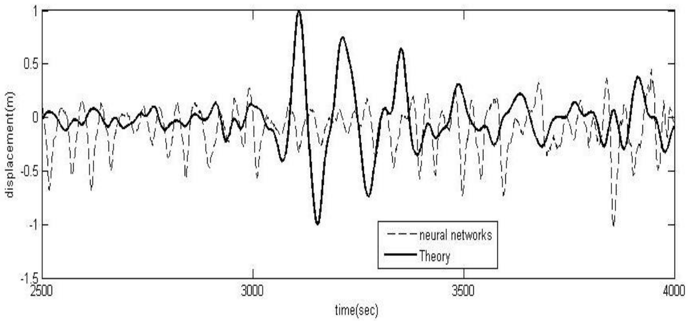



Figure 20. A=2 with B=13. Train with type1 and test with type2

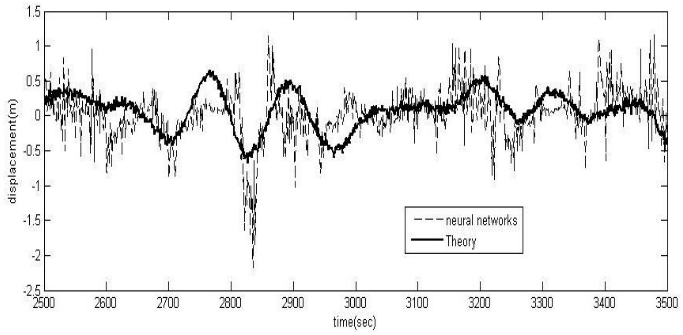



Figure 21. A=2 with B=13. Train with type1 and test with type3

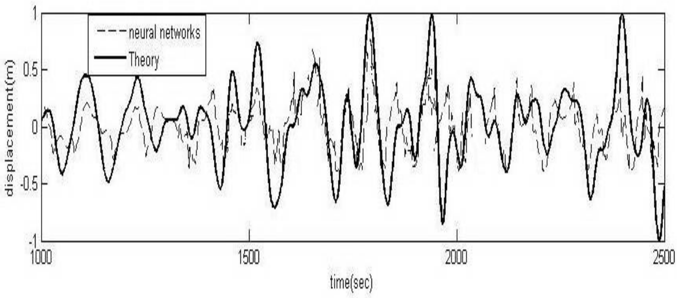



Figure 22. A=1 with B=4. Train with type1 and test with another type1

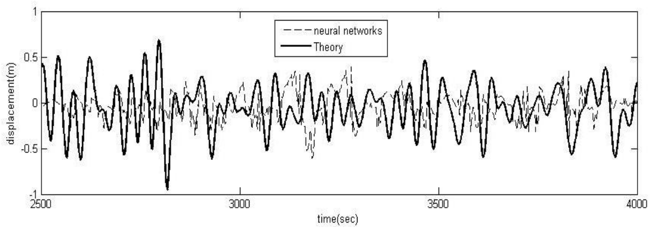



Figure 23. A=1 with B=4. Train with type1 and test with type2

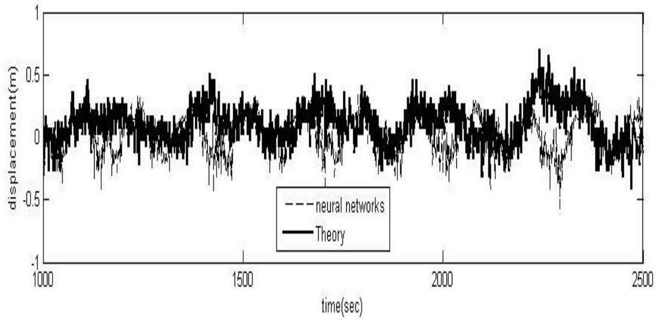



Figure 24. A=1 with B=4. Train with type1 and test with type3

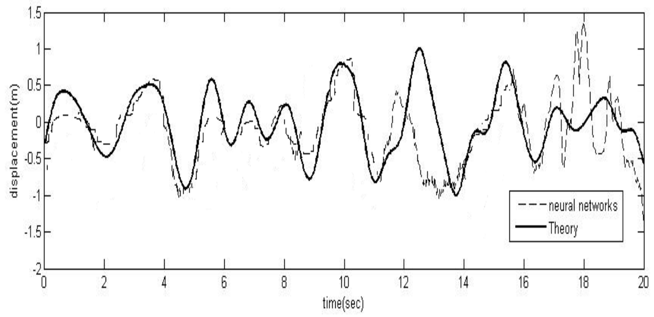



Figure 25. A=10 with B=12. Train with type1 and test with another type1

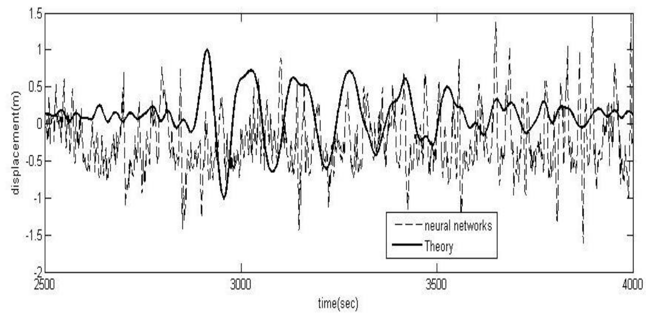



Figure 26. A=10 with B=12. Train with type1 and test with type2

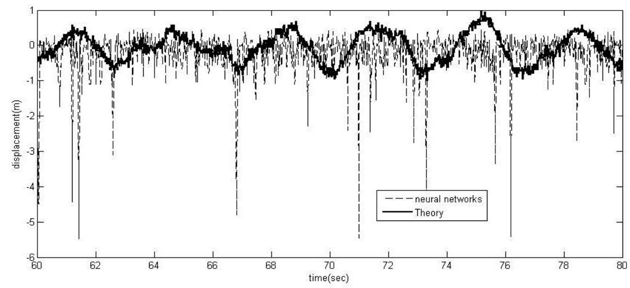



Figure 27. A=10 with B=12. Train with type1 and test with type3

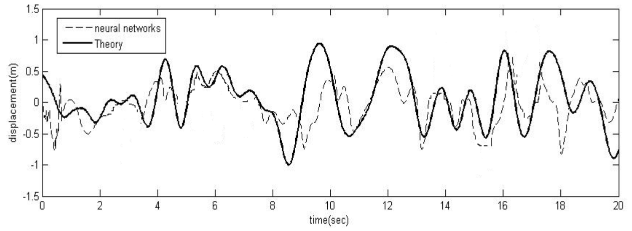



Figure 28. A=11 with B=8. Train with type1 and test with another type1

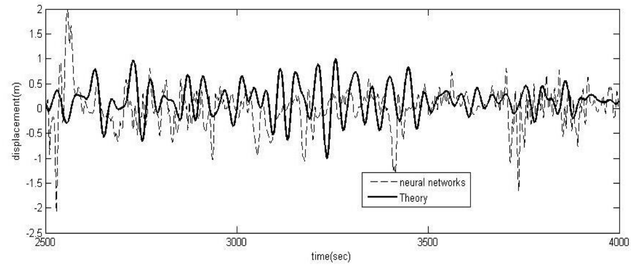



Figure 29. A=11 with B=8. Train with type1 and test with type2

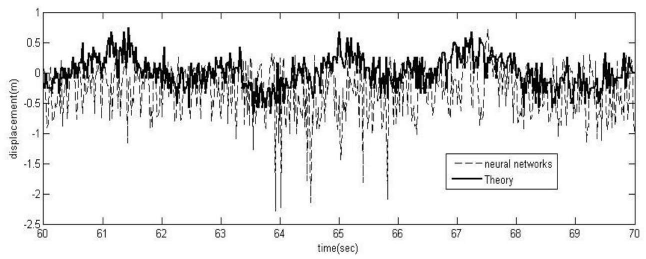



Figure 30. A=11 with B=8. Train with type1 and test with type3

## 3. CONCLUSIONS

The method of using neural network as explained above, has been successful in distinguishing the records corresponding to energy release due to induced cracks within the concrete wall from the records corresponding to environmental noise and from other sources. However, it is early to consider this technique as a proven technique for identification of data types. More work is needed.

## ACKNOWLEDGEMENT

Presentation of this paper was made possible by a support from the offices of higher education and research of Sharif University of Technology, Tehran, Iran. This support is greatly acknowledged.

## REFERENCES

[1] Hornik, K. (1991). “Approximation capabilities of multilayer feedforward networks.” Neural Networks, 4(2), 251–257.

[2] Joghataie, A., and Shafiei Dizaji, M. (2011). “Transforming results from model to prototype of concrete gravity dams using neural networks.” J. Eng. Mech., 10.1061/(ASCE)EM.1943-7889.0000246, 484–496.

[3] Yun, G. J., Ghaboussi, J., and Elnashai, A. S. (2008). “A new neural network-based model for hysteretic behavior of materials.” Int. J. Numer. methods Eng., 73(4), 447–469.

[4] Joghataie, A., and Shafiei Dizaji, M. (2013). “Designing high precision fast nonlinear dam neuro-modellers and comparison with finite elements analysis.” J. Eng. Mech., 10.1061/(ASCE)EM.1943-7889.0000572, 1311–1324.

[5] Masri, S. F., Chassiakos, A. G., and Caughy, T. H. (1933). “Identification of nonlinear dynamic systems using neural networks.” J. Appl. Mech., 60(1), 123–133.

[6] Joghataie, A. and Shafiei Dizaji, M. (2016). "Neuroplasticity in Dynamic Neural Networks Comprised of Neurons Attached to Adaptive Base Plate", Neural Networks, 75, 77-83.

[7] Farrokh, M. and Shafiei Dizaji, M. (2016). "Adaptive Simulation of Hysteresis Using Neuro-Madelung Model", Journal of Intelligent Material Systems and Structures, 27(13), 1713-1724.

[8] Siegelmann, H. T., Horne, B. G., and Giles, C. L. (1997). “Computational capabilities or recurrent NARX neural networks.” IEEE Trans. Syst. Man Cybernet. Part B, 27(2), 208–215.

[9] Farrokh, M., Shafiei Dizaji, M. and Joghataie A. (2015). "Modeling hysteretic deteriorating behavior using Generalized Prandtl neural network", ASCE, Journal of Engineering Mechanics, 141(8).

[10] Joghataie, A. and Shafiei Dizaji, M. (2009). "Nonlinear Analysis of Concrete Gravity Dams by Neural Networks", Proc. World Congress on Engineering, 1-3 July, (Ao, S. I., Gelman, L., Unter, A. and Korsunsky, A. AM., Editors), 1022-1027, London, England.