



SIMULATION OF DESIGN EARTHQUAKE IN THE HIMALAYAN REGION USING ARTIFICIAL NEURAL NETWORK

Jayanta PATHAK¹

D. K. PAUL²

P. N. GODBOLE³

SUMMARY

In engineering approach, it is necessary to simulate or select the design earthquake as spectrum compatible accelerograms for detailed dynamic analysis of structures. This work presents an Artificial Neural Network [ANN] based model to simulate spectrum compatible accelerograms as design earthquake. In a two-stage approach to simulate accelerograms from target spectra, modular compression was used to compress the Fourier spectra. The optimal architecture of the neural network to compress the Fourier spectra is determined by training them as replicator network to achieve a high speed of compression with reasonable compression ratio. Accelerogram generator neural networks are trained to inversely map the compressed vector of Fourier spectra to their corresponding response spectra. The methodology is extended to develop multiple accelerograms compatible with target spectra. More than hundred accelerograms recorded in the Himalayan region during ten Indian earthquakes, pre-classified based on their observed predominant frequencies, are used to train and test multiple neural networks. The networks are able to generate spectrum compatible accelerograms from target response spectra in all the categories of different predominant frequency content. The networks are also tested with smoothed design spectra as inputs and are able to synthesize ensemble of realistic accelerograms with desired frequency content.

INTRODUCTION

The basic objective of engineering approach is to simulate accelerograms as design earthquake, which can be used for analysis, evaluation, design and strengthening of structures for future hypothesized earthquakes. The practice of using response spectra has been popular with engineers, where seismic threat is postulated in terms of smoothed spectral shape. This practice has led to the development of various empirical spectrum compatible models, where ground motions are indirectly modeled to match amplitude, frequency content, duration with existing data. The indirectly modeled accelerogram from response spectra can be used for non-linear dynamic analysis employing time history analysis. The generation of

¹ Asstt. Professor, Civil Engineering Department,
Jorhat Engineering College, Assam, India, Email : jayanta_pathak@rediffmail.com

² Professor, Earthquake Engineering Department, IIT Roorkee, India

³ Ex-Professor, Civil Engineering Department, IIT Roorkee, India

accelerograms from response spectra is an inverse problem which does not have a unique solution. The inherent capability of ANN to learn inverse mapping from examples was first exploited by Ghaboussi and Lin (1998) to develop a new method for generating spectrum compatible accelerograms, which was later extended to generate multiple accelerograms from a given response spectrum by pre-classifying the accelerograms based on duration and using probabilistic neural network with stochastic neurons [Lin and Ghaboussi (2000), Lin (1999)].

The approach adopted in the present work is a variation of the methodology proposed by Ghaboussi and Lin (1998). More than hundred accelerograms recorded in the Himalayan region during ten Indian earthquakes have been utilized to train and test the proposed ANN based spectrum compatible model. The accelerograms used for training are pre-classified into categories based on their observed predominant frequencies. Multiple neural networks have been trained to map the accelerograms of each category to their corresponding response spectra to generate multiple accelerograms of different frequency content from given response spectra.

TYPE OF NETWORKS USED IN THE ANN BASED MODEL

There are essentially two different types of networks used in the ANN based spectrum compatible model for simulation of accelerograms. The first network is a data compression network, which compresses the FFTs of accelerograms from high dimensional vector to much lower dimensional vector. The data compression network is implemented as Replicator Neural Network (RNN) to avoid loss of information during compression. The second network is Accelerogram Generator Neural Network (AGNN), which trains the lower dimensional vectors representing the compressed form of the FFTs with their corresponding response spectra for generation of spectrum compatible accelerograms [Ghaboussi and Lin (1998)]. The implementation of RNN is necessary to reduce computational time while training AGNN.

Replicator Neural Networks (RNN) for Signal or Data Compression

Signal or data compression is used to achieve a low bit rate in the digital representation of signals with minimum loss of signal quality. The function of compression is usually referred to as low bit rate coding or simply coding. Signal compression has found wide application in many aspects of signal encryption, storage and communication [Tzou *et al.* (1994), Zhang *et al.* (1995), Jayant (1992)]. However, despite the efforts of many investigators working in this field, a good compromise involving quality, complexity and compression ratio has not yet been reached [Fa-Long and Rolf (1997)]. Recently there has been tremendous interest in applying ANN to signal compression and some promising results have been reported [Stark *et al.* (1991), Niemann *et al.* (1993), Dony *et al.* (1995)]. The characteristics of ANN involved in signal compression include massively parallel architecture; a high degree of interconnection; the propensity for storing experiential knowledge; and the capabilities of high speed computation, non-linear mapping, and self-organization [Dony *et al.* (1995)]. The high-speed computational capability of ANN can be employed to implement real time signal compression. The first serious studies of RNN for signal/data compression were carried out by Kohonen *et al.* (1977). Ackeley, Hinton and Sejnowski (1985) later studied the replicator neural networks in the context of the 'encoder problem'. Cortell *et al.* (1987) developed a replicator network version of the multi-layer feed-forward networks. Hecht-Nielson (1995, 1996) presented some theoretical studies clarifying certain fundamental aspects of the working of RNN. The present work has implemented RNN for data compression and decompression in two stages. There are two modules of RNN, which are RNN_1 and RNN_2. The first module RNN_1 is a two hidden layer network where the input and output layer each has 2049 nodes for compressing the real part of FFTs and there are m number of nodes in each of the hidden layer ($m \ll 2049$) giving a network architecture $2049-m-m-2049$. The network architecture of RNN_1 for imaginary part is $2047-m-m-2047$. The second module RNN_2 is a single hidden layer network where input and the output layer each has m number of

nodes and there are k number of nodes in the middle hidden layer ($k < m$) giving a network architecture $m-k-m$. A schematic representation of module RNN_1 and RNN_2 are given in Fig. 1.

Two separate RNN_1 (2049- $m-m$ -2049 and 2047- $m-m$ -2047) were trained to replicate the real and imaginary part of the FFTs respectively and once they were successfully trained for all the presented patterns of real and imaginary part of FFTs, the set of activation vectors of the first and second hidden layers were stored. The pair of activation vectors $[rA1_m]^p$; $[rA2_m]^p$ and $[iA1_m]^p$; $[iA2_m]^p$ represent the compressed signatures of real and imaginary part of FFTs, respectively.

In the second stage of compression two separate but identical RNN_2 ($m-k-m$) were trained for both real and imaginary part. The input to the input layer of RNN_2 trained for real part was the vector $[rA1_m]^p$ and the target output in the output layer was the vector $[rA2_m]^p$. Similarly, the input to the RNN_2 trained for imaginary part was the vector $[iA1_m]^p$ and the target output was the vector $[iA2_m]^p$. The lower part of Fig. 1 shows the schematic representation of RNN_2 trained for real and imaginary part of FFTs respectively. The module RNN_2 is not exactly a replicator neural network, but this module learns to map the pair of activation vectors given as input and target output respectively. Once this module learns to successfully map all components of the pair of activation vectors, it produces a further reduced low dimensional vector in its middle hidden layer having k number of nodes. These low dimensional vectors are $[rAC_k]^p$ and $[iAC_k]^p$ for the real and imaginary part of FFTs respectively and encode the compressed signatures of FFTs. The final replicator neural networks (RNN) having architectures 2049- $m-k-m$ -2049 and 2047- $m-k-m$ -2047 for compressing real and imaginary part of FFTs respectively are constructed by embedding the module RNN_2 into the module RNN_1 as shown in Fig. 2. The procedure of embedding RNN_2 to RNN_1 involves replacement of weight matrix $[rW_{hh}]$ and $[iW_{hh}]$ of networks in module RNN_1 by pair of weight matrix $[rw_1]$; $[rw_2]$ and $[iw_1]$; $[iw_2]$ of networks in module RNN_2 respectively.

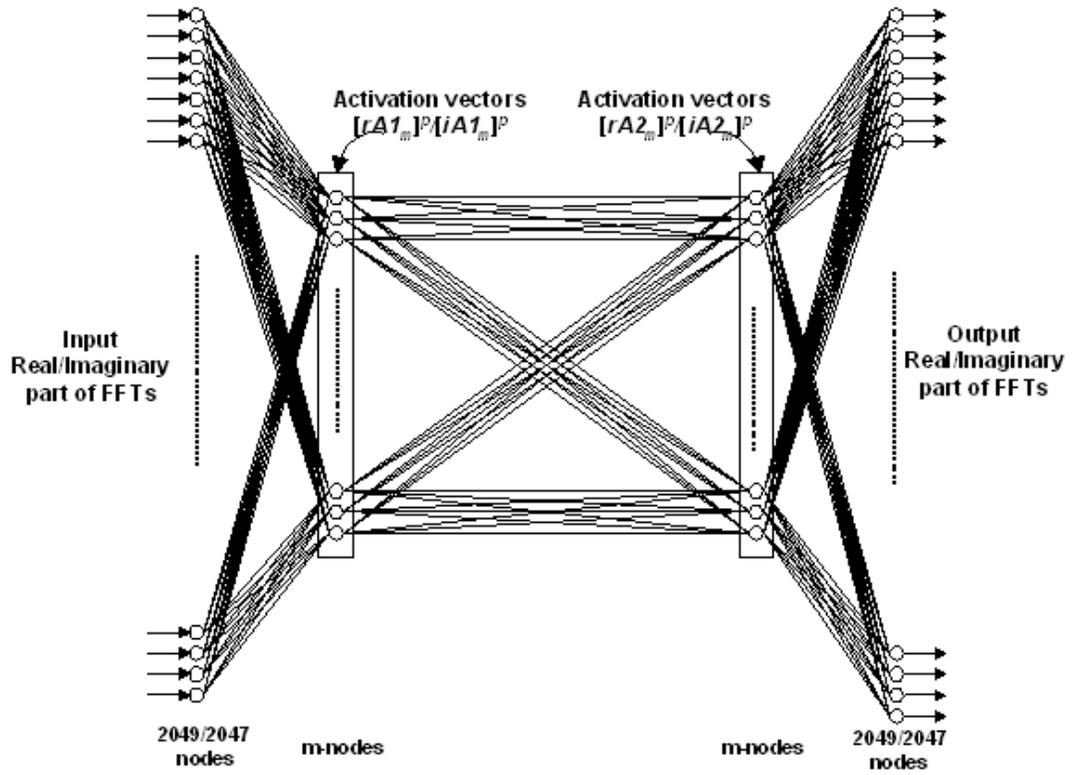
This phase wise compression was adopted to speed up the parametric study for finding optimum value of m and k so that an optimal configuration of RNN can be achieved having high speed of compression with reasonable compression ratio.

Parametric Study for Optimum Architecture of RNN

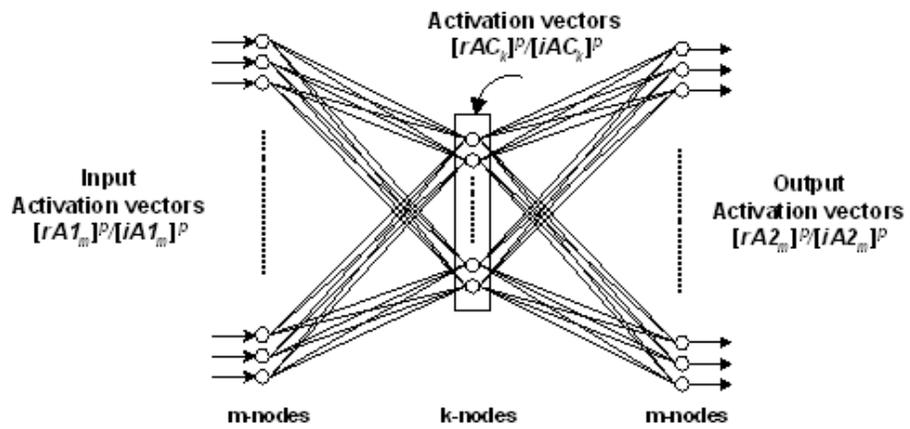
The parametric study for finding the optimum architecture of the RNN aims at finding the optimal number of nodes m and k in the outer hidden layers and middle hidden layer respectively for 38 randomly sampled recorded accelerograms from the data base. Out of these 38 recorded accelerograms, 8 accelerograms were kept aside for testing the networks.

The first module RNN_1 was trained to replicate the real and imaginary part of the FFTs by varying the number of nodes in the hidden layers (m). The Sum Square Error (SSE) after training various architectures of RNN_1 for 10000 epochs were compared. It was observed that the RNN_1 having number of hidden nodes less than 80 were not able to internally represent the both real and imaginary part of the FFTs used for training and were not converging well on the data set presented to the networks for replication. Therefore a value of $m < 80$ was ruled out for further parametric study to be carried out.

A series of network architectures for RNN_1 were then considered for training the data set by increasing the number of hidden nodes (m) from 80 to subsequent higher values. It was observed that the RNN_1 having 82 and 105 nodes in their hidden layers were consistently showing better convergence than other networks considered in the study while the networks were trained for successively lowered error level. Both the networks learned to replicate the real and imaginary part of the FFTs of all the 30 accelerograms at an error level of SSE=0.1.



Architecture of RNN_1



Architecture of RNN_2

Fig. 1 Schematic representation of module RNN_1 and RNN_2

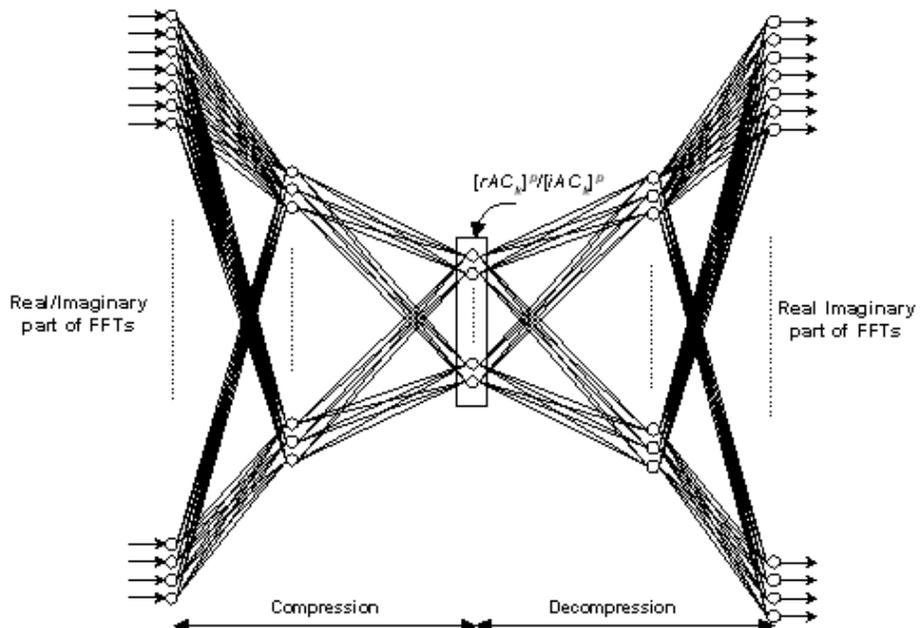
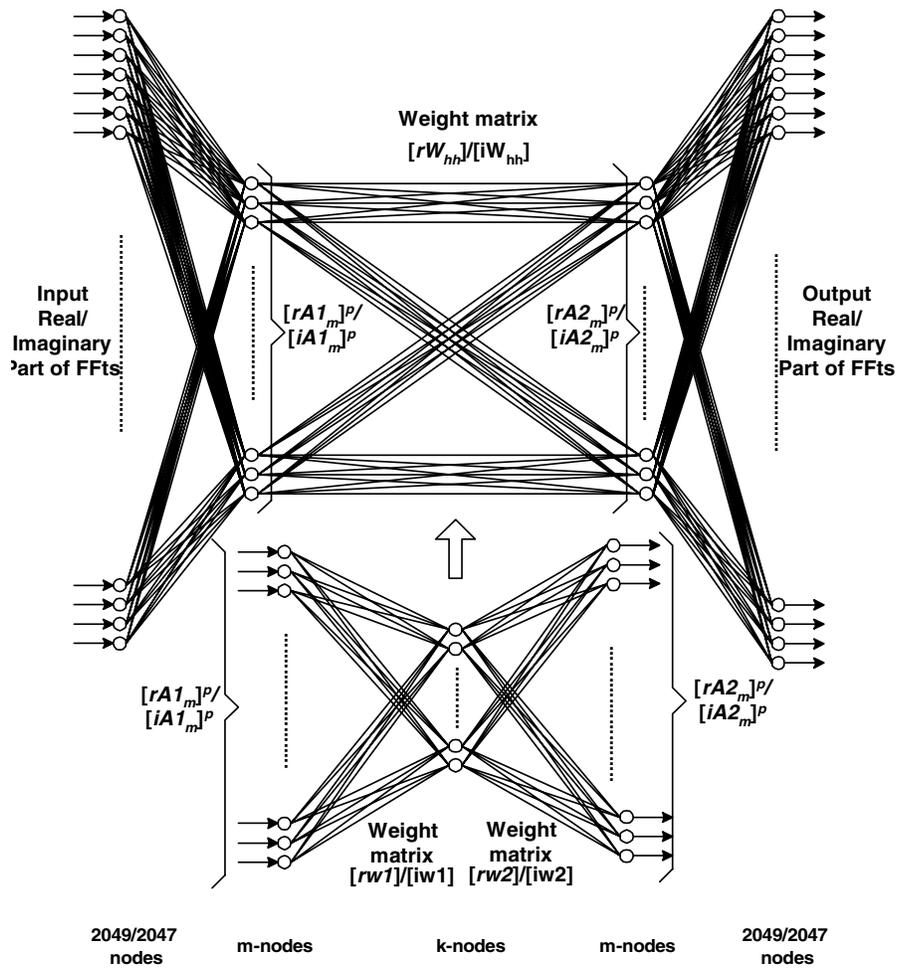


Fig. 2 Construction of RNN by embedding RNN_2 to RNN_1

Even though the number of epochs taken by the RNN_1 with 105 hidden nodes were lesser than the number of epochs taken by RNN_1 with 82 hidden nodes to replicate both real and imaginary part of FFTs, the CPU time taken by 2049-105-105-2049; 2047-105-105-2047 networks were higher than the 2049-82-82-2049; 2047-82-82-2047 networks, which was due to larger number of connections involved in the 2049-105-105-2049; 2047-105-105-2047 networks. However both the network architectures were retained for further evaluation of their performances.

The parametric study in the second stage of compression was carried out to find out the number of nodes k in the middle hidden layer of RNN_2. Two separate architectures of RNN_2 were considered at this stage, which were $82-k-82$ and $105-k-105$. The RNN_2 were trained to map the set of activation vectors $[rA1_m]^p$; $[rA2_m]^p$ and $[iA1_m]^p$; $[iA2_m]^p$ derived from the corresponding RNN_1 trained in the first stage of compression. The number of nodes m in each of the hidden layers of RNN_1 in the first stage of compression has a bearing on the optimal speed of compression. The number of nodes k evaluated in the second stage of compression decides the compression ratio that can be achieved for the final replicator neural network. Initially the value of k was fixed at 20 and a target error level was 0.005 (SSE). It was necessary to lower the target error level in the second stage of compression, so that the average error level at the output nodes of RNN_2 remain in the same order as that of RNN_1. It was observed that both the network architectures $82-20-82$ and $105-20-105$ were not able to converge to the desired error level while trained to map the corresponding hidden layer activation vectors for real and imaginary part of the FFTs. The value of k was subsequently increased and training was repeated. The networks finally converged for $k = 25$. Once, the networks $82-25-82$ and $105-25-105$ in module RNN_2 were successfully trained to map the set of activation vectors $[rA1_m]^p$; $[rA2_m]^p$ and $[iA1_m]^p$; $[iA2_m]^p$, they were embedded to the corresponding 2049-82-82-2049 and 2049-105-105-2049 networks in the module RNN_1 to construct the five layer RNN having architectures 2049-82-25-82-2049; 2047-82-25-82-2047 and 2049-105-25-105-2049; 2047-105-25-105-2047 respectively. It was observed that the two architectures of RNN were performing well on the training data set considered for the study, but their performances were evaluated on FFTs of 8 recorded accelerograms kept aside for testing.

The pseudo-velocity response spectra of the replicated time histories from the RNN having the two different architectures were compared with the pseudo-velocity response spectrum of the corresponding recorded accelerograms. The pseudo-velocity response spectra were calculated within the range of 0.01 and 50 Hz using Newmark Beta method with $\beta = 0.25$ and 5% damping. It was observed that the RNN having architecture 2049-82-25-82-2049; 2047-82-25-82-2047 were performing better than the RNN 2049-105-25-105-2049; 2047-105-25-105-2047, while tested on FFTs of novel accelerograms. The network architecture 2049-82-25-82-2049; 2047-82-25-82-2047 were finally selected as the optimum architecture of RNN for compressing the real and imaginary part of FFTs respectively for subsequent studies in the present work.

Performance of Replicator Neural Network

The trained replicator neural networks were tested by comparing replicated accelerograms with input accelerograms from training set. These comparisons were made for all the accelerograms included on the training set. It was observed that the RNN was exactly replicating the accelerogram at its output nodes and the response spectra of replicated accelerogram was exactly matching with the response spectra of recorded accelerogram. The RNN was also tested on novel accelerograms, which were not included in the training data set. It was observed that the RNN was able to replicate the novel accelerograms with reasonable accuracy and the response spectra of the replicated accelerograms were closely matching with the response spectra of recorded accelerograms. It was observed that the RNN was able to learn the

internal representation, while compressing the presented accelerograms and replicating them by decompression.

ACCELEROGRAM GENERATOR NEURAL NETWORKS (AGNN)

There are two identical AGNN, which are trained to relate the real and imaginary part of compressed FFTs of accelerograms with the pseudo-velocity response spectra at discrete frequencies separately. The first half of the AGNN is a double hidden layer network relating the pseudo-velocity response spectra to their corresponding compressed FFT signatures and the second half of AGNN is the decompression part of already trained RNN (2049-82-25-82-2049; 2047-82-25-82-2047).

The input layer of the first half of AGNN has 100 nodes, where they receive the values of the pseudo-velocity response spectra at 100 discrete frequencies. The pseudo-velocity response spectra were calculated within the range of 0.01 and 100 Hz using Newmark Beta method with $\beta = 0.25$ and 5% damping. The first half of the AGNN is trained to map the compressed signatures of FFTs with the pseudo-velocity response spectra. The training of AGNN involves the training of the connections of the first half of the AGNN and the weights of the connections in the second half of the AGNN are already trained and frozen weights of the decompression part of RNN. The architectures of the AGNN for generating real part of FFT from response spectra is shown in Fig. 6. The number of hidden nodes in the two hidden layers of the first half of AGNN were determined based on convergence criteria and a double hidden layer network with 100-32-32-25 architecture was found to be converging well up to a target error level of SSE=0.005.

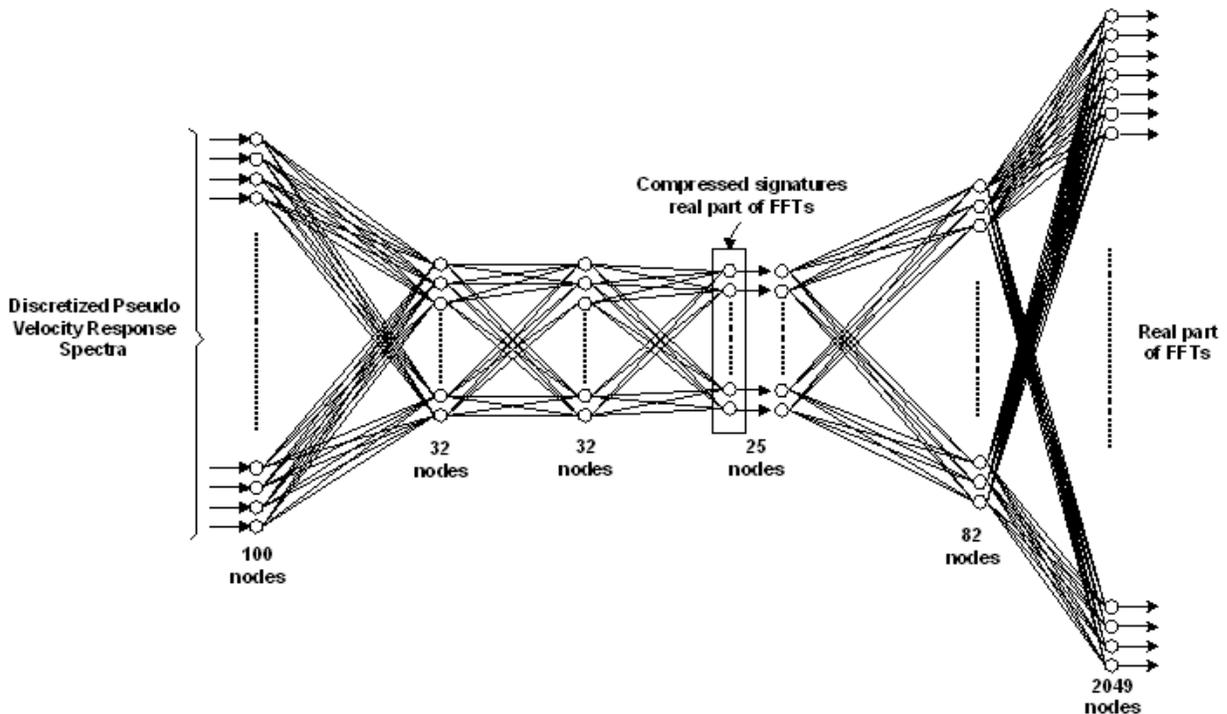
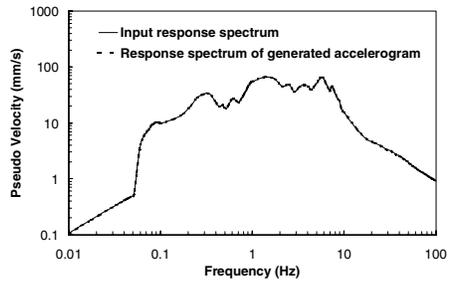


Fig . 3 Architecture of AGNN simulating real part of FFT



↑
AGNN
└

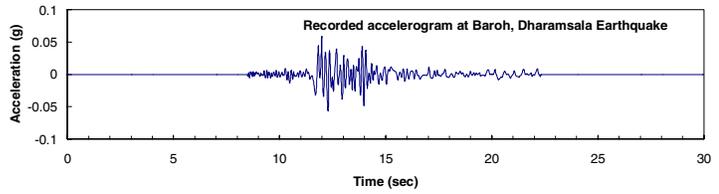
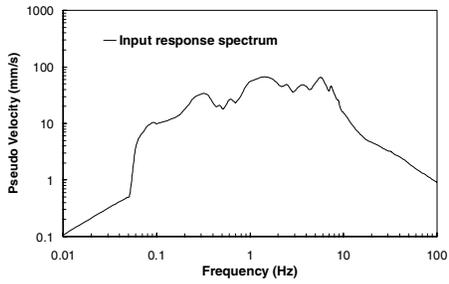
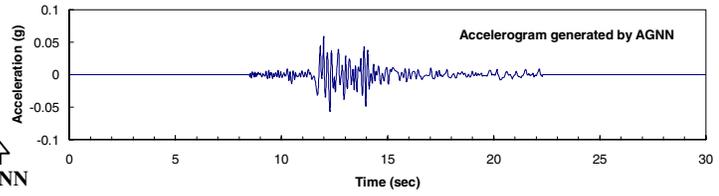
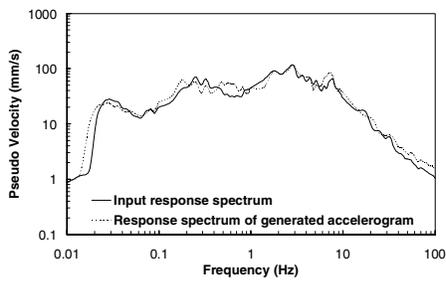


Fig. 4 Test of trained AGNN with response spectrum from the training set, Baroh, Dharamsala Earthquake



↑
AGNN
└

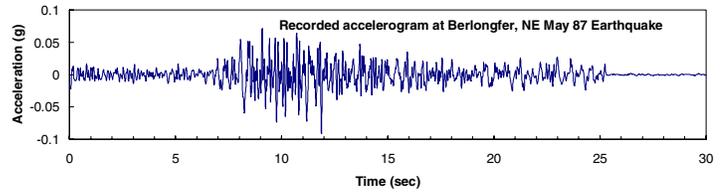
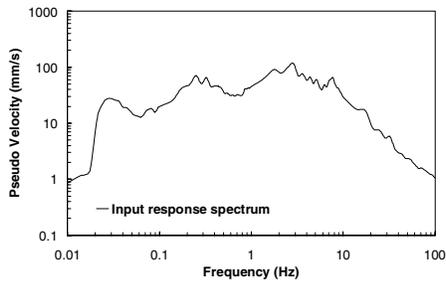
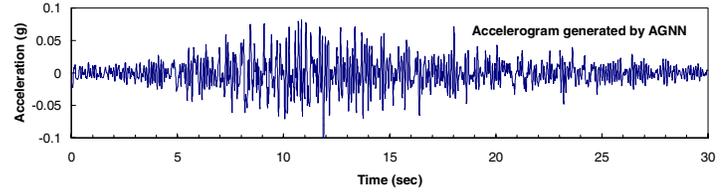


Fig. 5 Test of trained AGNN with response spectrum from the testing set, Berlongfer, NE May 87 Earthquake

Performance of Accelerogram Generated Neural Network

The trained AGNN was tested by presenting response spectra of accelerograms from training set and as well as novel response spectra from testing set. It was observed that, AGNN was able to generate the accelerograms accurately from the corresponding response spectra used for training. One example accelerogram generated by AGNN from response spectra used in training is shown in Fig. 4 for the Dharamsala earthquake.

The AGNN was also tested with novel response spectra, which were not included in the training data set. Figure 5 shows the generated accelerogram from input response spectrum at Berlongfer for the NE May 87 earthquake which was not included in training. It was observed that the AGNN generated an accelerogram from the novel input response spectrum and response spectrum of generated accelerogram matched closely with the input response spectrum. The above case study with the sample accelerograms established the feasibility of developing an ANN based model for generation of spectrum compatible accelerograms from given target spectra, using data recorded in the Himalayan region.

GENERATION OF MULTIPLE ACCELEROGRAMS FROM RESPONSE SPECTRA

Following the case study, efforts were made to generate multiple accelerograms from given response spectra. The frequency content of ground motion is one of the most important parameter as far as damage potential is concerned. A structure should be evaluated or analyzed for possible damage by subjecting it to either recorded or artificially generated ground motions having different predominant frequency contents so that the structure remains safe against possible earthquakes having energy in a wide range of frequency band. Moreover, it is always not possible to have a single recorded accelerogram which matches with the given target spectra or design spectra in all the ranges of frequency as design spectra are generally mean or average of several response spectra derived from more than one recorded accelerogram. The current practice adopted by structural engineers is to use multiple recorded accelerograms or artificially generated accelerograms which match with the given target spectra or design spectra in different frequency ranges. This work has attempted to generate multiple accelerograms from given target spectra or design spectra by pre-classifying over hundred recorded accelerograms based on their observed predominant frequency contents and training multiple AGNN to generate accelerograms having varying predominant frequency content. The Fourier amplitude spectrum of each recorded accelerogram was studied to derive the predominant frequency content of the accelerogram. The recorded accelerograms were classified into five categories namely very low frequency (< 1.5 Hz), low frequency (1.5 - 3.0 Hz), medium frequency (3 - 4.5 Hz), high frequency (4.5 - 6.0 Hz) and very high frequency (> 6.5 Hz) range. There were five different AGNN, which were trained and tested separately on these five categories of accelerograms, the results of which are presented and discussed in the following sections.

Generation of Accelerograms in various Frequency Ranges

The FFTs of the accelerograms considered for training were first compressed by RNN and the compressed signature of FFT were inversely mapped to the corresponding pseudo-velocity response spectra ($\beta=0.25$, 5% damping). Two identical AGNN_1 for very low frequency range were trained for both real and imaginary parts of the FFT. The weights of the connections of the decompression part of the RNN were kept frozen while training AGNN_1. An architecture of 100-21-21-25 for first half of AGNN_1 was found appropriate for training the accelerograms as this was the minimum possible network which learned to inversely map the compressed real and imaginary part of FFTs to their corresponding response spectra. The trained AGNN_1 was tested with response spectra from the training set and it generated the accelerograms, the response spectra of which were exactly matching with the input response spectra. The trained AGNN_1 was then tested with novel response spectra from the testing data set. It was observed that AGNN_1 was able generate accelerograms, the response spectra of which were closely matching with the input response spectra.

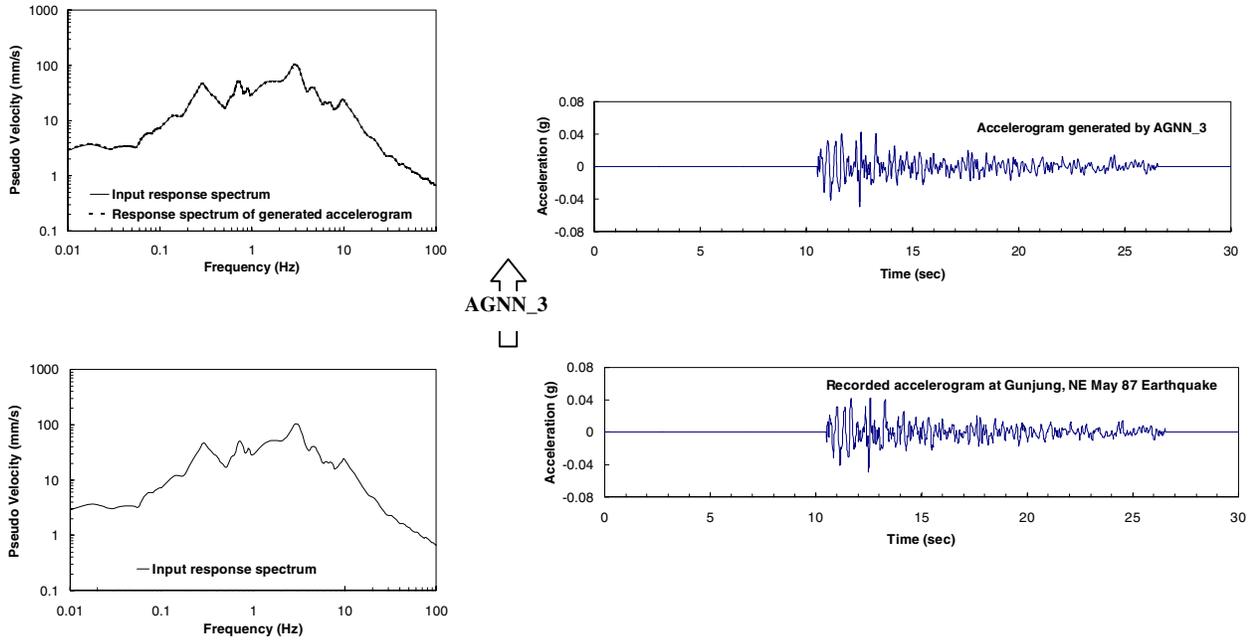


Fig. 6 Test of trained AGNN_3 with response spectrum from the training set, Gunjung, NE May 87 Earthquake

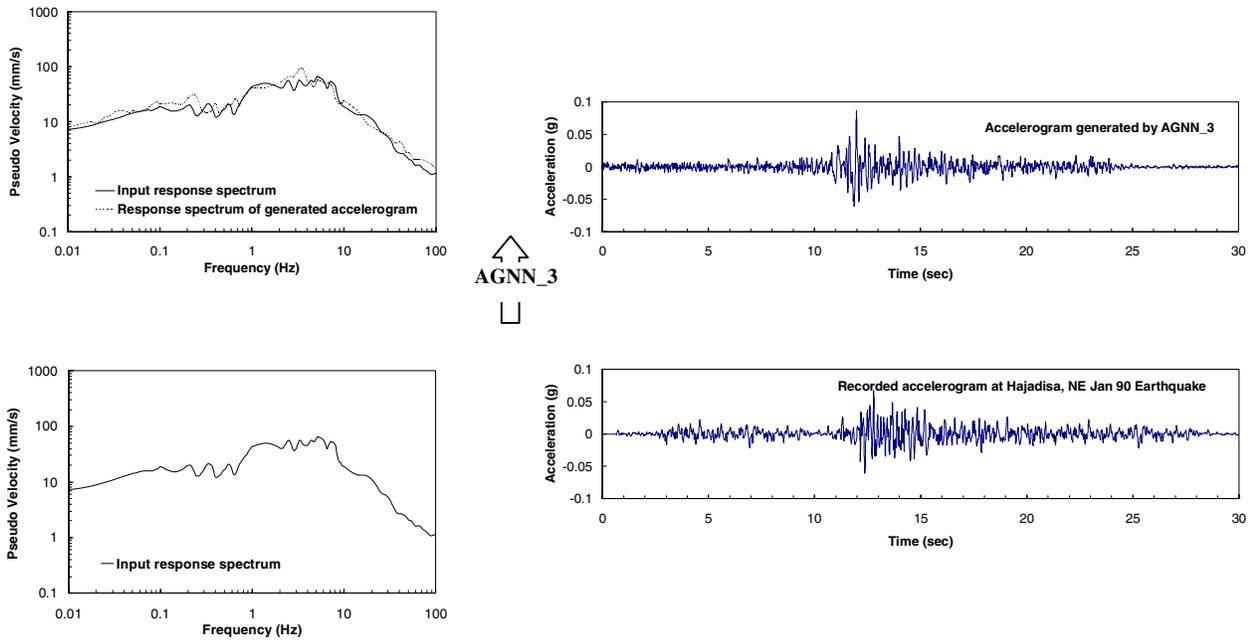


Fig. 7 Test of trained AGNN_3 with response spectrum from the testing set, Hajadisa, NE Jan 90 Earthquake

Similarly, an architecture of 100-27-27-25 for first half of AGNN_2 was found appropriate for training the accelerograms in low frequency range as this was the minimum possible network which learned to inversely map the compressed real and imaginary part of FFTs to their corresponding response spectra. In medium frequency range an architecture of 100-26-26-25 for first half of AGNN_3 was found appropriate for training the accelerograms. The trained AGNN_3 was tested with response spectra from the training set and it generated the accelerograms, the response spectra of which were exactly matching with the input response spectra as shown in Fig. 6. The trained AGNN_3 was then tested with novel response spectra from the testing data set. It was observed that AGNN_3 was able generate accelerograms, the response spectra of which were closely matching with the input response spectra as shown in Fig. 7. Following the same approach an architecture of 100-24-24-25 for first half of AGNN_4 was found appropriate for training the accelerograms in high frequency range and an architecture of 100-26-26-25 for first half of AGNN_5 was found appropriate for training the accelerograms in very high frequency range.

GENERATION OF ACCELEROGRAMS FROM DESIGN SPECTRA

It is interesting to determine whether the trained neural networks are capable of generating realistic accelerograms from design spectra, even though they have been trained with actual recorded earthquake accelerograms. Therefore, a case study was conducted by presenting design spectra as target spectra to the trained networks. This part of the study was performed with the full understanding that design spectra are usually the envelope of the response spectra of several accelerograms or their mean plus some multiple of standard deviation and not the response spectrum of a single accelerogram.

The design spectra used in the study are the pseudo-velocity response spectra derived from average acceleration response spectra proposed in Indian code IS : 1893-2002 Criteria for Earthquake Resistant Design of Structures [IS:1893(2002)]. The pseudo-velocity response spectra derived for two soil conditions namely rock or hard soil and soft soil are considered for the case study. The two pseudo-velocity response spectra were given as input to the different accelerogram generator neural networks (AGNN_1, AGNN_2, AGNN_3, AGNN_4 and AGNN_5) which are trained to generate accelerograms of different predominant frequency contents. It was observed that all the AGNN trained for various frequency ranges simulate spectrum compatible accelerograms in their desired frequency ranges. Figure 8 shows the accelerogram generated by AGNN_4 from pseudo-velocity response spectrum for rock or hard soil (5% damping). The upper part of the figure shows the comparison of response spectrum of generated accelerogram with input design spectrum. It was observed that AGNN_4 generated a realistic accelerogram the response spectrum of which closely matched with the input design spectrum. It was also observed that the accelerogram generated by AGNN_4 had predominantly high frequency as classified and trained by the network which is shown by the Fourier amplitude spectrum of the generated accelerogram in the upper part of the figure. The AGNN_4 is therefore generating realistic accelerograms in the desired predominant frequency. It should be mentioned here that even though the networks were never trained with design spectra as input, they were able to generate accelerograms with desired predominant frequency content.

It should be mentioned here that even though the networks were never trained with design spectra as input, they were able to generate accelerograms with desired predominant frequency content. This is a useful property of the neural network based methodology, which enable the networks to generate accelerograms compatible with specified design spectra or target spectra. This study shows the ability of the ANN based model to generate multiple accelerograms from given target spectrum as input. Each of the generated accelerograms are having varying predominant frequency content.

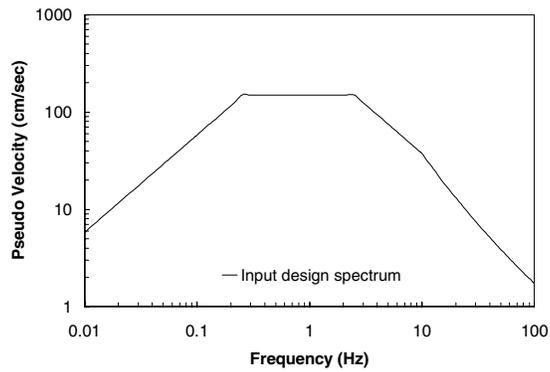
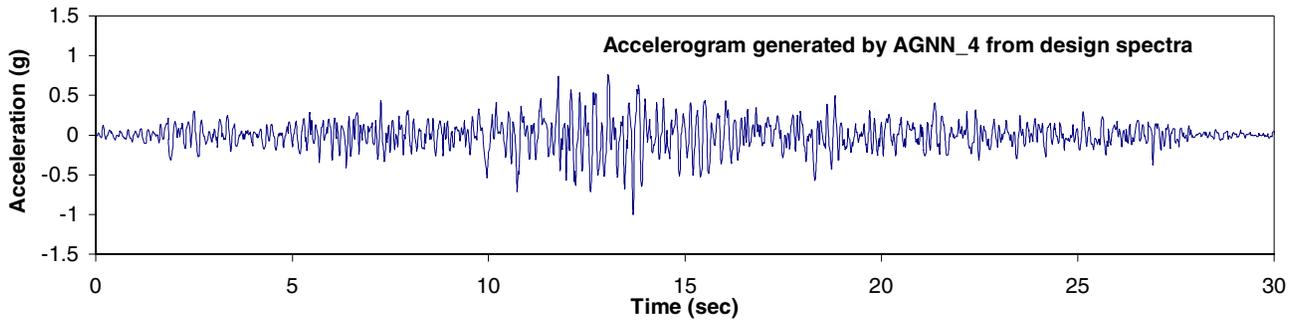
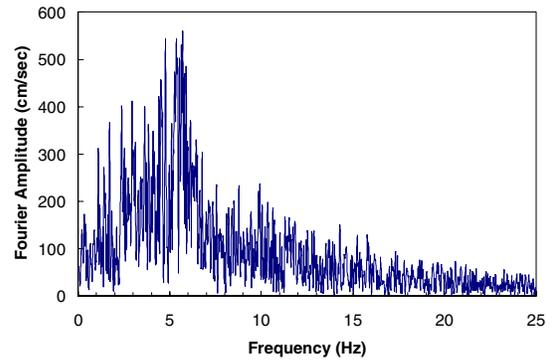
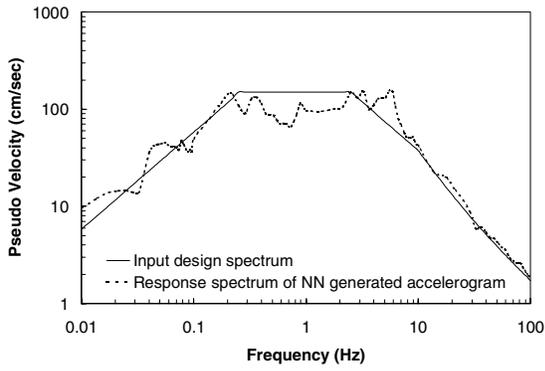


Fig. 8 Example of simulated Accelerogram by AGNN_4 From Design Spectrum Specified by Indian Code Is:1893-2002 For Rocky Or Hard Soil Sites

CONCLUSIONS

In this paper, results and discussions regarding an ANN based model for generation of accelerograms using data from strong-motion arrays in India is presented.

The replicator neural networks were used for compressing high dimensional FFTs to much lower dimensional vector. A modular two-stage compression was carried out for finding the optimal architecture of replicator neural network in terms of compression speed with a reasonable compression ratio. The performance of the replicator neural networks were tested for accelerograms used in training and as well as for novel accelerograms not used in training. Accelerogram generator neural networks were trained and tested on 38 randomly sampled accelerograms and it was observed that the recorded accelerograms in the Indian region can be utilized to develop a feasible inverse model for generating accelerograms from response spectra.

In the second phase of the study, neural networks were trained and tested to generate multiple accelerograms from target spectra. The recorded accelerograms used in this work were categorized based on their observed predominant frequencies and multiple networks were trained and tested for different categories. It was observed that the multiple networks trained on accelerograms categorized to five different categories learned the inverse mapping of accelerograms to their corresponding response spectra. The performance of the networks were tested by presenting novel response spectra as input and it was observed that the networks learned to generate accelerograms from response spectra in all the categories of different frequency ranges.

The ANN based model was finally tested by presenting pseudo-velocity response spectra derived from average acceleration response spectra specified in Indian Standard code IS : 1893 as input and it was observed that the model was able to generate multiple realistic accelerograms in different frequency ranges.

REFERENCES

1. Ackeley, D. H., Hinton, G. E. and Sejnowski, T. J. "A Learning Algorithm for Boltzmann Machines." *Cognitive Science*, 9, 1985, 147-169.
2. Code IS:1893-2002. Indian Standard Criteria for Earthquake-Resistant Design of Structures, New Delhi, India, 2002.
3. Cottrell, G. W., Munro, P. and Ziper, D. "Learning Internal Representations From Gray-Scale Images: An Example of Extensional Programming." *Proc. Ninth Annual Conference, Cognitive Science Society, Seattle, WA*, 1987.
4. Dony, R. D. and Haykin, S. "Neural Network Approaches to Image Compression." *Proc. IEEE*, 83(2), 1995, pp. 288-303.
5. Fa-Long and Rolf. "Applied Neural Network for Signal Processing." Cambridge University Press, UK, 1997.
6. Ghaboussi, J. and Lin, C.-C. J. "New Method of Generating Spectrum Compatible Accelerograms Using Neural Network." *Earthquake Engng. Struct. Dyn.*, 27(4), 1998, 377-396.
7. Hecht-Nielson, R. "Replicator Neural Networks for Universal Optimal Source Coding." *Science*, 269, 1995, 1860-1863.
8. Hecht-Nielson, R. Data Manifolds, "Natural Coordinates, Replicator Neural Networks, and Optimal Source Coding." *ICONIP-96*, 1996.
9. Jayant, N. Signal "Compression Technology Targets and Research Directions." *IEEE J. Selected Areas in Communications*, 10(5), 1992, 796-818.
10. Kohonen, T., Lehtiö, P., Rovamo, J., Hyvärinen, J., Bry, K. and Vainio, L. "A Principle of Neural Associative Memory." *Neuroscience*, 2, 1977, 1065-1076.
11. Lin, C.-C. "A Neural Network Based Methodology for Generating Spectrum Compatible Earthquake Accelerograms." Ph.D. Thesis, Department of Civil and Environmental Engineering, University of

Illinois At Urbana-Champaign, Urbana, Illinois, 1999.

12. Lin, C.-C. J. and Ghaboussi, J. "Recent Progress on Neural Network Based Methodology for Generating Artificial Earthquake Accelerograms." Proc. 12th World Conference on Earthquake Engineering, Auckland, New Zealand, 2000.
13. Niemann, H. and Wu, J. K. "Neural Network Adaptive Image Coding." IEEE Trans. on Neural Networks, 4(4), 1993, pp. 615-27.
14. Stark, J. A "Neural Network to Compute the Hutchinson Metric in Fractal Image Processing", IEEE Trans. on Neural Networks, 2(1), 1991, 156-58.
15. Tzou, K. H., Musmann, H. G. and Aizawa, K. (Eds.). Special Issue on "Very-Low-Bit-Rate Video Coding". Proc. IEEE Trans. on Circuits and Systems for Video Technology, 4(3), 1994.
16. Zhang, Y. Q, Li, W. and Liou, M. L. (Eds.). Special Issue on "Advances in Image and Video Compression". Proc. IEEE, 83(2), 1995.