A Quantitative Comparison Of Three Design Algorithms for Feedforward Neural Networks

Adriana Dumitras and Adrian Traian Murgan
Electronics and Telecommunications Faculty
"POLITEHNICA" University of Bucharest
1–3, Iuliu Maniu Blvd.,
77206 Bucharest
ROMANIA

Abstract: A comparison of three pruning algorithms for feedforward neural networks design is proposed, namely MOBD (Modified Optimal Brain Damage), TOBD (Tridiagonal Optimal Brain Damage) and OBD (Optimal Brain Damage). The application taken into account is nonlinear regression and the evaluation criteria are: mean square error, peak signal–to–noise ratio, total number of parameters in the final neural structure, weight values, number of "multiply + add" operations in the test phase and required test time.

Keywords: Feedforward neural network, pruning method, saliency, symmetry

Adriana Dumitras Ph.D, currently she is senior assistant professor, at the Applied Electronics Department, "Politehnica" University of Bucharest. She has published 26 scientific papers on topics in the fields of artificial neural networks and computer architectures. She was a recipient of the 1993 TEMPUS grant for carrying out a research period at the Technical University of Denmark.

Adrian Traian Murgan Ph.D, Professor and Head of the Applied Electronics Department, "Politehnica" University of Bucharest. He has published more than 100 scientific papers on information theory, neural networks and fuzzy systems. He was a recipient of the "Alexander von Humboldt" Stiftung grant. He was visiting professor at several German universities. He is a member of IEEE.

1. Introduction

As powerful nonlinear signal processing tools, neural networks have been extensively applied in pattern recognition, classification, filtering and estimation, etc. Attempts to optimize the neural network structures have been made, in order to avoid overfitting and improve generalization, to obtain higher convergence speed and less costly implementations.

There are several techniques for optimizing the neural architecture [1, 2]. Basically, they are: empirical methods, methods based on statistical criteria [1, 2, and others], growing (constructive) [7–13, and others], decreasing (destructive, pruning) [14–20, and others] and hybrid methods [21–24]. Reviews of the growing [3–5,] and decreasing [6] techniques are also available.

Largely discussed during the last few years [6, 14–20, and others], mainly with applications in time series prediction, pruning methods lead to compact networks, which show good performance as compared to the starting architecture or to other structures of greater size. Though the resulting configuration is sparsely connected, usually this is not symmetrical. In hardware implementations, or even in software simulations, the designer is sometimes deeply frustrated, due to the lack of symmetry: the weight storage, the programs written for digital signal processors, etc., would benefit from the symmetry of the weight matrices. Many times in circuit theory, symmetry and reciprocity conditions are imposed on circuits. It is clear that a symmetrical neural network structure would be useful. This is the reason why, in the following, we shall make a quantitative comparison of pruning algorithms with symmetry constraints, for feedforward neural networks.

2. Theoretical Approach

A survey in pruning algorithms [6] quotes two broad classes of pruning methods: those which estimate the sensitivity of the error function to the removal of an element and secondly, methods which add terms to the objective function that rewards the network for choosing efficient solutions. There is some overlap of these two groups, since the objective functions could include sensitivity terms. The destructive algorithms evaluated in the following, belong to the former group, including also symmetry constraints [25–27].

2.1 The Optimal Brain Damage Algorithm

In the Optimal Brain Damage Algorithm (OBD) method, the feedforward neural network (FANN) is trained and the weight saliencies are calculated. The weights with the lowest saliencies are eliminated and finally, the network is retrained.

The weight's saliency is defined as the change in the training error when the weight is eliminated and the remaining weights are retrained to the new minimum [17, 18].

Basically, the saliency is approximated by the second derivative of the cost function w.r.t. the weight.

The OBD technique was applied to pruning FANNs in contiguity problems, time series prediction, etc. [17, 18].

2.2 The Modified Optimal Brain Damage Algorithm

[25, 26] proposed, instead of a simple elimination as in standard OBD, a symmetric pruning of the weights. In the Modified Optimal Brain Damage Algorithm (MOBD), at each step, the weight having the lowest saliency is eliminated, together with its symmetric (as position in the FANN) "pair".

MOBD led to sparse weight matrices, which contained zero values in symmetrical positions. The expense of getting symmetrical networks was an increase in the test error [25, 26].

2.3 The Tridiagonal Optimal Brain Damage Algorithm

Another solution of symmetric pruning in feedforward neural networks, the Tridiagonal Optimal Brain Damage Algorithm (TOBD), was suggested in [26, 27]. Without loss of generality, one assumes that the weight matrix has the weight vectors as columns. After calculating the saliencies, the column containing the minimum saliency values is determined and a Householder (reflection) transform is performed on the weight matrix. The weights are normalised and a retraining step follows. After several steps, symmetrical weight values resulted in a tridiagonal matrix. Details of the algorithm and experimental results have been presented in [26, 27].

3. Experimental Results

3.1 The Neural Network Architecture and Learning Algorithm

Let the problem be nonlinear regression and let the FANN be a M–H–N multilayer perceptron (M input, H hidden and N output nodes), trained with the backpropagation algorithm with momentum. We have chosen $M = 7$, $H = 3$, $N = 1$.

The training data consisted of 2880 samples of the "chirp" signal (Figure 1), fed into the network through a 7–sample window, sliding one step to the right. The desired output value was the next value following the window (i.e. the eighth). The training set was learned in 2000 epochs, with a learning rate of 0.01 and a momentum equal to 0.0001. The mean square learning error after 2000 epochs, normalised to the number of patterns was 0.004456.

The test data (Figure 1) consisted of 2880 patterns, different from the training data.

3.2 Evaluation Criteria

The evaluation criteria taken into account are: mean square error scaled to the number of patterns (SMSE) in both training and test phases, peak signal–to–noise ratio (PSNR), total number of parameters (weight, biases) in the final neural structure, weight values, the number of "multiply + add" operations in the

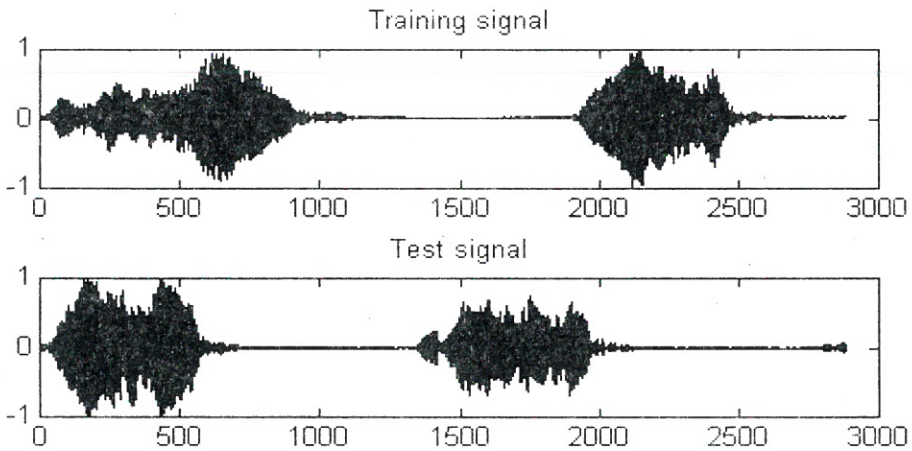


Figure 1. Training and Test "Chirp" Signals

test phase and the required test time.

The SMSE cost function is given by (3), where $y_j(\xi)$ and $d_j(\xi)$ are the actual, and respectively, the desired output values, with $1 \le j \le N$, for each input pattern $1 \le \xi \le P$. One denotes by $e_j(\xi)$ the output error (i.e. the difference $d_j(\xi) - y_j(\xi)$) in node j for the input pattern ξ and by $\mathbf{w}$, the FANN parameters.

$$C(\mathbf{w}) = \frac{1}{N\,P}\sum_{\xi=1}^{P} C(\mathbf{w};\xi) =$$

$$= \frac{1}{N\,P}\sum_{\xi=1}^{P}\sum_{j=1}^{N} C\left(\mathbf{w}; e_j(\xi)\right) =$$

$$= \frac{1}{N\,P}\sum_{\xi=1}^{P}\sum_{j=1}^{N}\left\{\frac{1}{2}\left[d_j(\xi) - y_j(\xi)\right]^2\right\} \quad (3)$$

The PSNR in decibels is given by (4), where d_{max} and d_{min} are the maximum and respectively, the minimum value of the desired signal.

3.3 Results of the Quantitative Evaluation

3.3.1 Learning and Test Errors

The MOBD, TOBD and OBD learning errors, test errors and PSNR after the pruning steps are

$$PSNR = 10\log_{10}\left[\frac{P\left(d_{max} - d_{min}\right)^2}{\sum_{\xi=1}^{P}\sum_{j=1}^{N}\left[d_j(\xi) - y_j(\xi)\right]^2}\right] =$$

$$= 10\log_{10}\left[\frac{\left(d_{max} - d_{min}\right)^2}{SMSE}\right] \quad (4)$$

shown in Table 2. The comparison was made after 500, 1,000, 2,000, 3,000 and 4,000 retraining epochs, corresponding to the pruning steps shown in Table 1. Both MOBD and TOBD have higher test errors as compared to OBD. However, the difference between MOBD and OBD is not significant, while for TOBD further retraining is necessary in order to increase the PSNR.

Learning curves during retraining for MOBD and OBD are shown in Figures 2 and 3, where the peaks point to the weight elimination steps. No significant differences may be noticed for the two algorithms.

The test errors evaluated with Akaike's Final Prediction Error (FPE) criterion are lower for all the algorithms taken into account than the errors yielded by the AR(7) and ARMAX(7) models (Table 3).

3.3.2 Peak Signal–to–Noise Ratio

Peak signal–to–noise ratio decreases as the number of the FANN parameters changes (Tables 2, 4). The highest values are given by the FANN in the OBD algorithm, closely followed by MOBD. However, PSNR evaluation after more than 10,000 epochs yields a higher value as compared to the fully connected network output (i.e. better generalization).

Table 1. Number of Retraining Epochs (NRE) and the Corresponding MOBD, TOBD and OBD Pruning Step

	NRE	MOBD	TOBD	OBD
Step	500	1	1	2
	1000	2	2	4
	2000	4	3	6
	3000	6	4	9
	4000	7	5	10

Table 2. Learning Errors, Test Errors and PSNR for MOBD, TOBD and OBD Algorithms

NE	MOBD			TOBD			OBD		
	LE	TE	PSNR [dB]	LE	TE	PSNR [dB]	LE	TE	PSNR [dB]
500	0.00332	1.05915	67.23	0.00437	1.07333	65.85	0.00332	1.05846	67.65
1000	0.00322	1.06813	67.23	0.00370	1.07341	62.40	0.00322	1.06818	64.95
2000	0.00394	1.06648	64.98	0.00307	1.07342	63.42	0.00325	0.82931	71.06
3000	0.00331	1.07159	64.84	0.00290	1.07341	64.63	0.00315	0.78907	71.60
4000	0.00277	1.06524	66.89	0.00308	1.07343	65.79	0.00339	0.98223	68.29

NRE = Number of retraining epochs; LE = Learning error; TE = Test error

Table 3. Test Errors Estimated With Akaike's FPE, for MOBD, TOBD, and OBD Algorithms, As Compared To AR(7) and ARMAX(7) Models

Method / model	Estimated test error, using FPE
MOBD	0.0028
TOBD	0.0030
OBD	0.0034
Fully connected net	0.0045
ARMAX(7)	0.0150

3.3.3 Total Number of Parameters in the Final Neural Network

The total number of network parameters given by MOBD, TOBD and OBD methods is shown in Table 5, and it is compared to the fully connected FANN. The lowest number of parameters is given by the TOBD algorithm, followed by MOBD and OBD. The final neural structures for MOBD, TOBD and OBD algorithms are shown in Figure 4.

Table 4. The PSNR Values, As the Number of the FANN Parameters (NP) Decreases (Dec. = Decreases, Inc. = Increases)

Method	PSNR	NP decreases:
MOBD	Dec. with 1.079 dB	1.65 times
TOBD	Dec. with 2.200 dB	3.11 times
OBD	Inc. with 0.290 dB	1.27 times

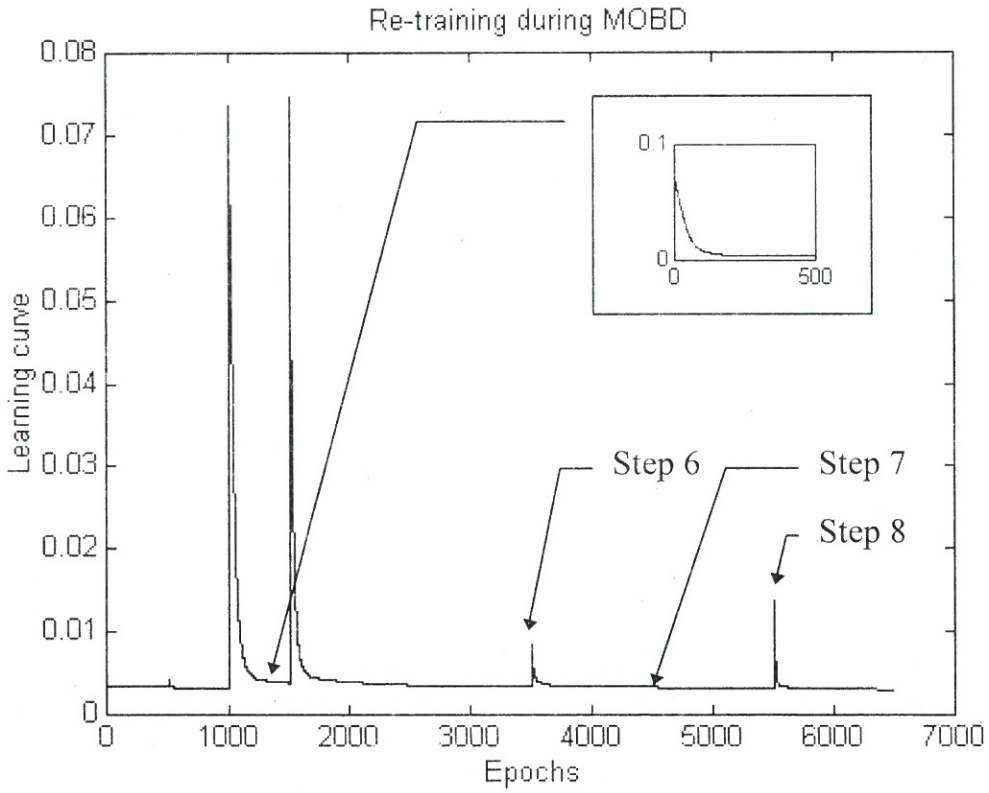


Figure 2. Retraining During MOBD. Peaks Show Weight Elimination Steps

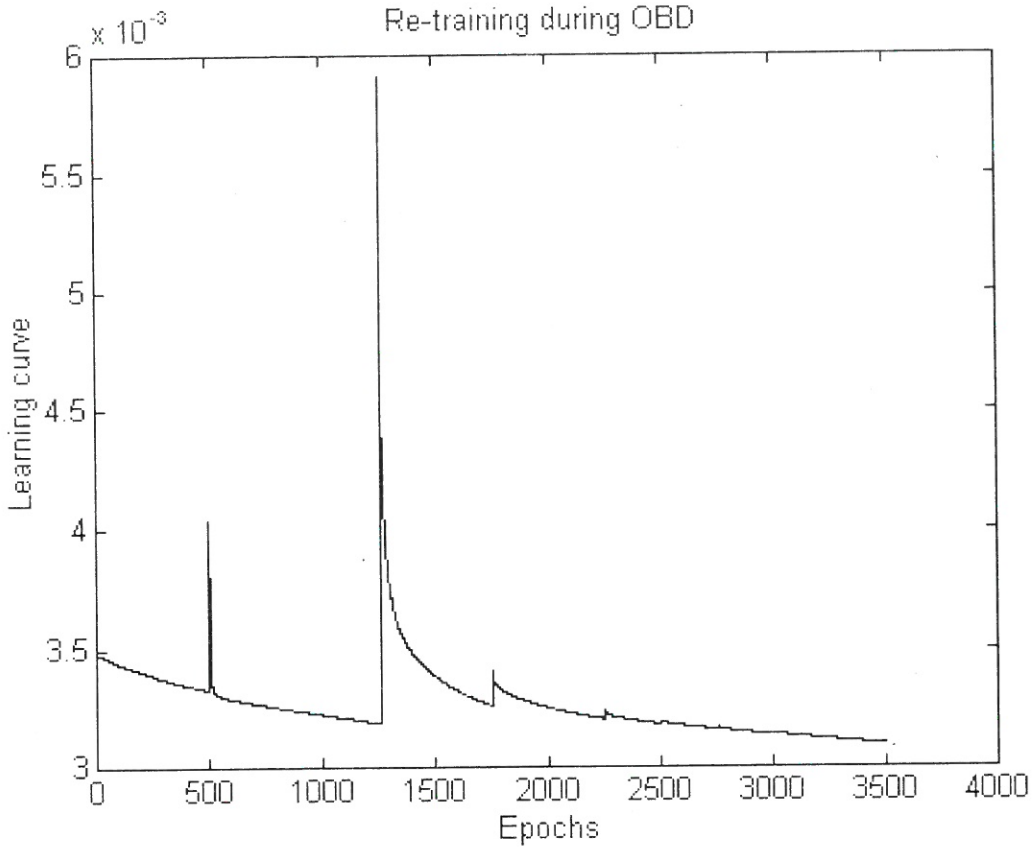
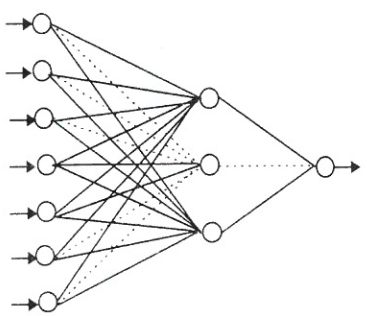
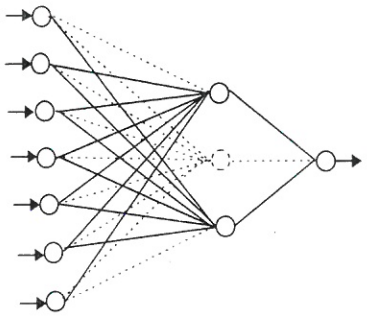


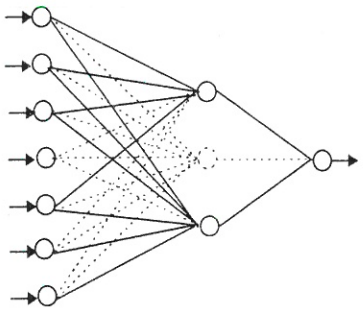
Figure 3. Retraining During OBD. Peaks Show Weight Elimination Steps



The network structure after OBD



The network structure after MOBD



The network structure after TOBD

Figure 4. Neural Network Structures Resulted From OBD, MOBD and TOBD
Dotted Lines Represent Eliminated Connections / Nodes

Table 5.The Number of Weights, Biases and Stored Parameters in TOBD, MOBD and OBD Algorithms

Algorithm	Weights	Biases	No. of stored parameters
TOBD	12	3	9
MOBD	14	3	17
OBD	18	4	22
Fully connected network	24	4	28

The FANN resulted from OBD may be further simplified, using more pruning steps. The MOBD structure is symmetrical, like the TOBD net, but in the latter case, only 7 of the 12 parameters have to be stored, as they have symmetrical values. The second hidden node was eliminated in all of the three methods.

3.3.4 Weight Values

The input – hidden and hidden – output weight values resulted from MOBD and OBD, are shown in Figures 5 and 6. The comparison was performed for the same number of retraining epochs. The hidden – output weights at each step in TOBD and OBD are shown in Figures 7 and 8. One may notice that there are no significant differences as to the range of values. However, in TOBD symmetrical connections have been preserved in the structure.

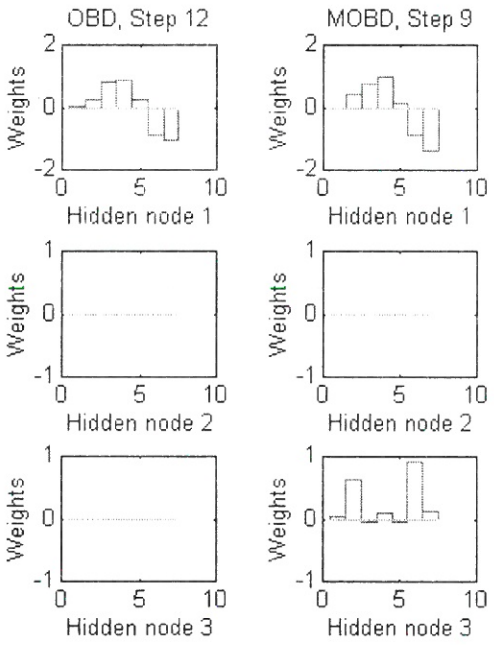


Figure 5. Comparison Between Input – Hidden Weight Values Resulted From MOBD and OBD

3.3.5 Number of "Multiply + Add" Operations in the Test Phase

The number of "multiply + add" operations in the test phase (Table 6) decreases 4.66 times for TOBD, 2.47 times for MOBD and 1.90 times for OBD structures, as compared to the number of operations in the fully connected network.

3.3.6 Required Test Time

Theoretical evaluations of the required test time [37, 38] and experimental values on an IBM – PC 486 / 66 MHz computer are included in Table 6. One denotes by t_0 the time required by a simple add or multiply operation and by t_t the transfer time for 16–bit data.

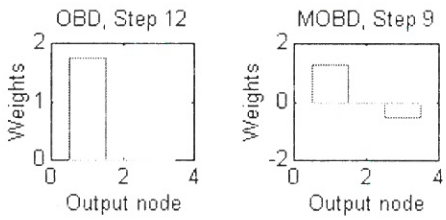


Figure 6.Comparison Between Hidden – Output Weight Values Resulted From MOBD and OBD

4. Conclusions and Future Work

We have performed a quantitative comparison of the MOBD, TOBD and OBD algorithms. The first and the second led to symmetrical, sparsely connected networks, while the third led to a non – symmetrical sparsely connected network.

The results show that the structure decreases dramatically in all cases. The lower mean square test error was reached in the OBD case, followed by MOBD and TOBD, but it corresponds to the highest number of parameters preserved in the neural structure.

Table 6. The number of "Multiply + Add" Operations in the Test Phase and the Required Test Time for the FANN Structures Resulted From MOBD, TOBD and OBD Algorithms

Neural network structure resulted from:	Number of "multiply + add" operations in the test phase	Required test time	
		Theoretical	Experim. [sec]
TOBD	14810	5760 ($5\ t_t + 7\ t_0$)	15
MOBD	27975	5760 ($6\ t_t + 7\ t_0$)	22
OBD	36205	5760 ($7\ t_t + 9\ t_0$)	25
Fully connected FANN	69120	5760 ($7\ t_t + 10\ t_0$)	42

Although the range of the weight values is basically the same, in the MOBD and TOBD methods there is no need to store all the final parameters, as some of them have symmetrical values. same hierarchy is maintained for the required test time.

The comparison performed in this paper leads to the conclusion that no symmetry may be introduced in the neural model, without paying

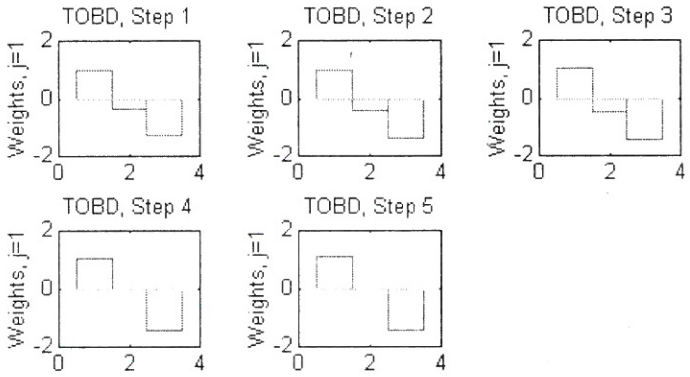


Figure 7. Hidden – Output Weight Values Resulted From TOBD

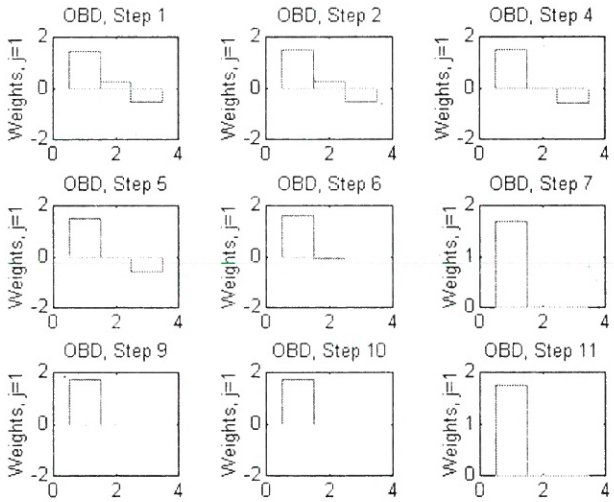


Figure 8. Hidden – Output Node Weight Values, After OBD Steps With Retraining

The number of "multiply + add" operations is minimum for the FANN given by the TOBD algorithm, followed by MOBD and OBD. The the price of increasing the error, i.e. decreasing the PSNR. Better performances are likely to obtain if one refines the approximations in the OBD algorithm.

Joining this matter of future research, there is another problem we currently focus on, regarding theoretical boundaries in using sparse weight matrices to approximate the FANN input – output relationship.

REFERENCES

1. CICHOCKI, A. and UNBEHAUEN, R., **Neural Networks for Optimization and Signal Processing**, JOHN WILEY & SONS, UK, 1993.
2. 2. HERTZ, J., KROGH, A. and PALMER, R. G., **Introduction To the Theory of Neural Computation**, ADDISON - WESLEY PUBL. CO, USA, 1991.
3. KWOK, T.-Y. AND YEUNG, D.-Y., – **Constructive Feedforward Neural Networks for Regression Problems: A Survey**, Technical Report HKUST–CS95–43, The Hong Kong University of Science and Technology, Department of Computer Science, September 1995.
4. HEN HU, Y., **Configuration of FeedForward Multilayer Perceptron Neural Networks**, Preprint, University of Wisconsin , Madison, Department of Electrical and Computer Engineering, USA, 1994.
5. FIESLER, E., **Comparative Bibliography of Ontogenic Neural Networks**, Proc. of ICANN'94: Intl. Conference on Artificial Neural Networks, Sorrento, Italy, 1994, pp. 793-796.
6. REED, R., **Pruning Algorithms– A Survey**, IEEE TRANSACTIONS ON NEURAL NETWORKS, Vol. 4, No. 5, September 1993, pp. 740-747.
7. FAHLMAN, S.E. and LEBIERE, C., **The Cascade – Correlation Learning Architecture**, Technical Report CMU–CS–90–100, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, August 1991.
8. HOEHFELD, M. and FAHLMAN, S.E., **Learning With Limited Numerical Precision Using the Cascade – Correlation Algorithm**, Technical Report CMU–CS–91–130, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, May 1991.
9. HORNIK, K., STINCHCOMBE, M. and WHITE. H., **Multi – Layered FeedForward Neural Networks Are Universal Approximators**, NEURAL NETWORKS, Vol. 2, Elsevier Science Ltd, USA, 1990, pp. 359-366.
10. FRATTALE MASCIOLI, F.M., MARTINELLI, G. and LAZZARO, G., **Comparison of Constructive Algorithms for Neural Networks**, Proc. of ICANN'94: Intl. Conference on Artificial Neural Networks, Sorrento, Italy, 1994, pp. 731-734.
11. MÉZARD, M. and NADAL, J. P., **Learning in Feedforward Layered Networks: The Tiling Algorithm**, JOURNAL OF PHYSICS, No. 22, USA, 1989, pp. 2191-2203.
12. NAHBAN, T.M. and ZOMAYA, A.Y., **Toward Generating Neural Network Structures for Function Approximation**, NEURAL NETWORKS, Vol. 7, No. 1, Elsevier Science Ltd, USA, 1994, pp.89-99.
13. REFENES, A.N. AND VITHLANI, S., **Constructive Learning by Specialization**, Proc. of the Intl. Conference on Neural Networks, Helsinki, Finland, 1991.
14. XUE, Q., HEN HU, Y. and TOMPKINS, W.J., **Structural Simplification of A FeedForward, Multi – layer Perceptron Artificial Neural Network**, Proc. of ICASSP'91, Toronto, Canada, 1991.
15. COTTRELL, M., GIRARD, B., GIRARD, Y., MANGEAS, M. and MULLER, C., **SSM: A Statistical Stepwise Method for Weight Elimination,** Proc. of ICANN'94: Intl. Conference on Artificial Neural Networks, Sorrento, Italy, 1994, pp. 601-604.
16. CIBIAS, T., SOULIÉ, F.F., GALLINARI, P. and RAUDYS, S., **Variable Selection With Optimal Cell Damage**, Proc. of ICANN'94: Intl. Conference on Artificial Neural Networks, Sorrento, Italy, 1994, pp. 327-330.
17. GORODKIN, J., HANSEN, L.K., SVARER, C. and WINTHER, O., **A Quantitative Study of Pruning By Optimal Brain Damage**, Preprint, Electronics Institute, Technical University of Denmark, Lyngby, Denmark, 1993.
18. HASSIBI, B. and STORK, D.G., **Second Order Derivatives for Network Pruning: Optimal Brain Surgeon**, in S.J. Hanson et al (Eds.) NIPS 5, San Matteo, CA, USA, MORGAN KAUFMANN PUBL. CO, 1993, p.164.

19. KUNG, S.Y. and HEN HU, Y., **A Frobenius Approximation Reduction Method (FARM) for Determining Optimal Number of Hidden Units**, Proc. of the IEEE Intl. Conference on Neural Networks, Seattle, WA, USA, July 1991, pp.163-168.

20. THODBERG, H.H., **Improving Generalization of Neural Networks Through Pruning**, INTL. JOURNAL OF NEURAL SYSTEMS, USA, Vol. 1, No. 4, 1991, pp. 317 – 326.

21. CHEN, Y. Q., THOMAS, D.W. AND NIXON, M.S., **Generating Shrinking Algorithm for Learning Arbitrary Classification**, NEURAL NETWORKS, Vol. 7, No. 9, Elsevier Science Ltd, USA, 1994, pp.1477-1489.

22. DUMITRAS, A., LAZARESCU, V. AND MURGAN, A.T., **A Growing – Decreasing Method for Designing Neural Filters**, in I. Pitas (Ed.) Proc. of the 1995 IEEE Workshop on Nonlinear Signal and Image Processing, Vol. 2, Neos Marmaras, Greece, 1995, pp. 579-582.

23. HIROSE, Y., YAMASHITA, K. and HIJIYA, S., **Back-propagation Algorithm Which Varies the Number of Hidden Units**, NEURAL NETWORKS, No. 4, USA, 1991, pp. 61-66.

24. HANSEN, L.K. and PEDERSEN, M. W., **Controlled Growth of Cascade Correlation Nets**, ICANN'94: Intl. Conference on Artificial Neural Networks, Sorrento, Italy, 1994, pp. 797-780.

25. DUMITRAS, A. et al, **MOBD: A Neural Network Pruning Algorithm with Symmetry Constraints**, ICSP'96: Intl. Conference on Signal Processing, Beijing, China, October 1996.

26. DUMITRAS, A., **Artificial Neural Network Structures for Digital Signal Processing**, Ph.D Thesis, Electronics and Telecommunications Department, "Politehnica" University of Bucharest, Romania, November 1996.

27. DUMITRAS, A. et al, in preparation.