Test Bed for Multilayered Feed forward Neural Network Architectures as Bidirectional Associative Memory

Manisha Singh
Assistant Professor, School of Engineering & Technology
Sharda University, Greater Noida, UP, India

ABSTRACT
Multilayered feed-forward neural networks are considered universal approximators and hence extensively been used for function approximation. Function approximation is an instance of *supervised learning* which is one of the most studied topics in machine learning, artificial neural networks, pattern recognition, and statistical curve fitting. Bidirectional associative memory is another class of networks which has been used for approximating various functions. In the present study, an approach for using MLFNN architectures as BAM with BP learning has been proposed and initially been tested on certain functions. The results obtained are analyzed and presented.

Keywords
Neural networks, Multilayered feed-forward neural network (MLFNN), Bidirectional Associative Memory (BAM), function approximation

1. INTRODUCTION
The adaptive systems are the ones which provide an optimal and robust solution subjected to a process called *learning*. The main advantage of the adaptive systems over the non-adaptive schemes lies in their self adjusting and time varying capabilities. We find the application of adaptive systems in a range of applications like prediction, function approximation etc. *Function approximation* is the task of learning or constructing a function that generates approximately the same outputs from input vectors as the process being modeled, based on available training data. The approximate mapping system should give an output which is fairly close to the values of the real function for inputs close to the current input used during learning. There exist multiple methods that have been established as function approximation tools, where an artificial neural network (ANNs) is the one which has been very popular recently. A trained neural network is expected to capture the system characteristics in their weights. Such network is supposed to have generalized from the training data, if for a new input the network produces the same output which the system would have produced. Function approximation is an instance of *supervised learning*, the primary topic studied in machine learning, artificial neural networks, pattern recognition, and statistical curve fitting. In principle, any of the methods studied in these fields can be used in reinforcement learning.

Multilayered feed-forward neural networks (MLFNNs) [1-2] and bidirectional associative memory (BAM) [3] are two types of ANN which have been extensively used for function approximation or pattern mapping tasks. MLFNNs generally use back-propagation (BP) algorithm for learning while in BAM architectures the learning is achieved through Hebbian rule. Due to design, MLFNN with hidden layers is used for the classification purposes and hence when used for pattern mapping task, suffers the problem of generalization.

2. LITERATURE REVIEW
Many authors including Cybenk (1989), Hecht-Nielsen (1989), Carroll and Dickinson (1989), Hornik (1990, 1993), Park and Sandberg (1991, 1993), Barron (1993) have studied function approximation using feed-forward neural networks. The structure studied varied in the terms of number of hidden layers for a particular function. It is an established fact that a two-layered FNN, i.e. one that does not have any hidden layers, is not capable of approximating generic nonlinear continuous functions (Widrow, 1990). On the other hand, five or more layer FNNs are rarely used in practice. Hence, almost all the work deals with the most challenging issue of the approximation capability of three/four-layered FNNs.

The multilayer feed-forward neural networks have simpler dynamics compared to the recurrent neural networks which is the generic class of BAM [4-6]. To increase the limited storage and mapping capability of BAMs substantial work in the literature has been reported [7]-[18]**.** This includes application of high-order nonlinearity to forward and backward information flows [7], application of an exponential scheme of information flow to exponentially enhance the similarity between an input pattern and it’s nearest stored pattern [8], high capacity fuzzy associative memory (FAM) for multiple rule storage [9], a weighted-pattern learning algorithm [9]. Other related work can be found in [10], [11],[12][13] in which either by adding dummy neuron, increasing in number of layers or manipulating the interconnection among neurons in each layer, the issue of performance improvement of BAM is addressed. Even some new learning algorithms were introduced to improve the performance of original BAM [14]-[18].

3. PROPOSED METHODOLOGY
The proposed method undertakes a MLFNN architecture and attempts to work it as BAM, with the help of application of BP algorithm in two passes. Firstly at the input node, the input to the network is provided and the output is calculated at the output layer, the difference between the actual output and the expected output (error) when propagated in the backward direction, adjusts weight to make the difference between both to a desired level. This is the standard application of BP algorithm. Now in the present approach, in the next pass, the output is provided at the output layer assuming it to be the input of BP and the standard BP is run in the reverse direction and again the weights are adjusted accordingly. These two passes together make a single run which let the MLFNN work as BAM. The schematic representation is given the figure and the detailed steps of algorithm are outlined in figure.

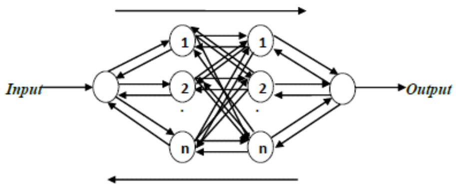

Fig 1: *A typical* ***1-n$_1$-n$_2$-1*** *feed-forward architecture as Bidirectional Associative Memory*

Pass-I

The standard BP algorithm

- *Create a feed-forward network with one input, n hidden units and one output.*
- *Initialize all the weights to small random values.*
- *Repeat until* $(E_{avg} = \sum_{k=1}^{Q} Ek < \tau)$ *where* τ *is the tolerance level of error*

1. *Input pattern* X_k *from training set and compute the corresponding output* S_y
2. *For corresponding output, calculate the error as*

$$\delta_o = S_y(D - S_y)(1 - S_y)$$

3. *For each hidden unit* h*, calculate*

$$\delta_h = S_h(1 - S_h) W_{hj} \delta_o$$

4. *Update each network weight between hidden and output layer* W_{hj} *as follows:*

$$W_{hj} = W_{hj} + \eta\Delta W_{hj} + \alpha\,\Delta W_{hj}old$$

where $\Delta W_{hj} = S_h * \delta_o$ *and* $\Delta W_{hj}old = \Delta W_{hj}$

5. *Update each network weight between input and hidden layer* W_{ih} *as follows:*

$$W_{ih} = W_{ih} + \eta\Delta W_{ih} + \alpha\,\Delta W_{ih}old$$

where $\Delta W_{ih} = \delta_h * X_k$ *and* $\Delta W_{ih}old = \Delta W_{ih}$

Pass-II

- *Repeat until* $(E_{avg} = \sum_{k=1}^{Q} Ek < \tau)$ *where* τ *is the tolerance level of error*

1. $W_{hi} = W'_{ih}$ *and* $W_{jh} = W'_{hj}$
2. *Input pattern* D_k *for corresponding* X_k *and compute the corresponding output* S_u
3. *For corresponding output* S_u*, calculate the error as*

$$\delta_i = S_u(X - S_u)(1 - S_u)$$

4. *For each hidden unit* h*, calculate*

$$\delta_h = S_h(1 - S_h) W_{hi} \delta_i$$

5. *Update each network weight between hidden and input layer* W_{hi} *as follows:*

$$W_{hi} = W_{hi} + \eta\Delta W_{hi} + \alpha\,\Delta W_{hi}old$$

where $\Delta W_{hi} = S_h * \delta_i$ *and* $\Delta W_{hi}old = \Delta W_{hi}$

6. *Update each network weight between output and hidden layer* W_{jh} *as follows:*

$$W_{jh} = \Delta W_{jh} + \eta\Delta W_{jh} + \alpha\,\Delta W_{jh}old$$

where $\Delta W_{jh} = \delta_i * D_k$ *and* $\Delta W_{jh}old = \Delta W_{jh}$

Fig 2: *Proposed Two-Pass BP algorithm*

4. SIMULATION OF EXPERIMENTS AND RESULTS

A four-layered feed-forward neural network has been created for the experimentation. Since the example function is univariate, therefore at the input and output layers only one neuron is present. Various combinations of neurons ranging from 2 to 20 in both hidden layers are experimented. The summary of the various parameters used is shown in **Table 1.**

At different learning rates and fixed momentum value (0.6), the convergence of the function has been studied with tolerance value .005. The standard BP algorithm and the proposed two-pass algorithm (shown in **figure 2**) were run one after other and the results of the epochs at which the convergence was achieved are compiled in **Table 2.** One epoch in standard BP algorithm means when all the training patterns are presented once, while one epoch in proposed two-phase BP algorithm means when for all patterns both forward and backward phases run once.

Table 1: *Various parameters used*

Parameter	***Number/Values***
Input Neurons	1
Hidden Neurons in hidden layer 1 ***(n1)***	2……20
Hidden Neurons in hidden layer 2 ***(n2)***	2……20
Number of Patterns	5
Learning rate, η	0.8/0.9/1.0
Momentum, α	0.6
Tolerance, τ	0.005

Table 2: *Number of epochs required for convergence for approximating the example function*

Selected Architecture *(input-*	*Learning rates*					
	η=0.8		***η=0.9***		***η=1.0***	
	FFN	BAM	FFN	BAM	FFN	BAM
1-2-2-1	2923	922	2331	728	2674	679
1-2-3-1	3813	541	2639	797	4142	614
1-2-4-1	3122	503	2138	711	2484	423
1-2-5-1	2815	615	2792	618	2470	433
1-3-2-1	2774	647	3940	629	2345	793
1-3-3-1	1822	530	1905	486	3047	557
1-3-4-1	2759	582	1735	432	2167	562
1-3-5-1	1938	560	1993	418	2550	416
1-4-2-1	4671	546	1706	580	2160	584
1-4-3-1	2670	646	2133	479	1770	412
1-4-4-1	2796	518	1968	521	2496	394
1-4-5-1	2161	478	2186	441	1353	369
1-5-2-1	2466	529	1805	498	1282	451
1-5-3-1	3000	629	1696	531	1349	520
1-5-4-1	1563	575	2395	445	1718	629
1-5-5-1	1481	525	1821	506	1490	459
1-5-10-1	1496	382	1733	298	1179	325
1-10-5-1	1356	390	1170	338	1231	290
1-10-10-1	971	319	1033	317	717	294
1-10-15-1	880	299	543	294	732	263
1-15-10-1	742	277	803	271	718	213
1-15-15-1	623	225	526	239	592	222
1-15-20-1	722	225	477	208	495	209
1-20-15-1	593	252	520	221	503	185
1-20-20-1	546	231	533	192	483	183

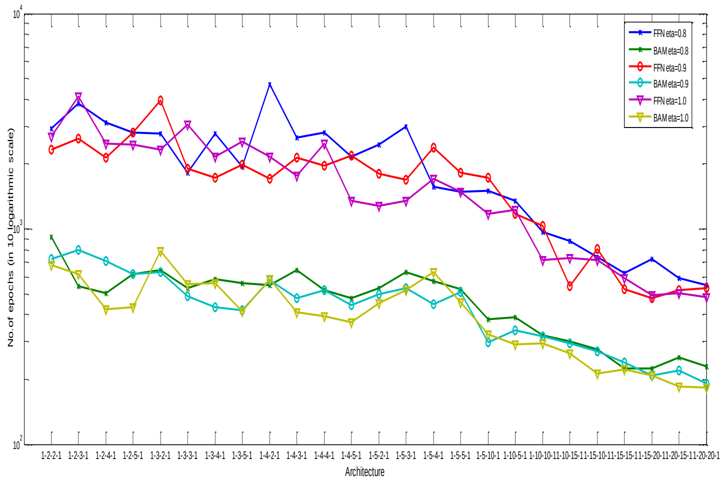

Fig 3: *Comparative graphs of the number of epochs required for convergence in MLFNNs and BAM for the various architectures*

The results shown in **Table 2** and **Figure 3** clearly indicate that the proposed two-pass algorithm for BAM architecture, in general, outperforms the MLFNN architecture with standard back propagation algorithm. The number of epochs taken for the convergence of the taken example function is substantially low in the case of proposed two-pass BP algorithm, in almost every architecture studied. It is observed that the number of epochs kept on lessening when we move on increasing the number of neurons in the hidden layers and various combinations are explored. Also there are some architectures in between where a larger number of epochs were recorded for convergence (e.g. 1-2-4-1, 1-4-2-1 etc.). Since both the algorithms work on the randomness of initial weight matrices, this may be one of the reasons for such results. Another obvious observation comes out that on increasing the learning rate from 0.8 to 1.0 the convergence is being achieved in lesser number of epochs, in general.

5. CONCLUSION AND FUTURE SCOPE

A new variation of back propagation algorithm, the Two-Pass BP, has been presented which when applied to multi-layered feed-forward neural network (MLFNN) let it behave like bidirectional associative memory (BAM). The four-layered architectures having various combinations of neurons in the two hidden layers have been experimented for a single valued-function. The results presented above suggest that proposed algorithm is better suited for function approximation and results obtained are quite encouraging. The work is still in early stage and need to be tested on larger number of architectures with more input patterns and different types of functions.

REFERENCES

[1] Rumelhan D.E., Hinton G.E., and Williams R. J**.,** Learning Internal Representations by Error Propagation, *Parallel Distributed Processing, Vol. I* Foundations, MIT Press, Cambridge, MA, pages 318-364. 1986.

[2] Tang C. Z**.** and Kwan H**.** K., Parameter effects on convergence speed and generalization capability of back-propagation algorithm *International Journal of Electronics***,** *74(1)*: 35-46, January 1993.

[3] Kosko B., *Neural Networks and Fuzzy Systems* (*Prentice*-Hall. Inc., New Jersey, 1992)

[4] Pearlmutter B.A., Gradient calculations for dynamic recurrent neural networks: a survey. *IEEE Trans. On Neural Networks***,** *6 (5):*1212-1228, Sept. 1995

[5] Prokhorov D.V., Feldkamp L.A., and Tyukin I.Y**.,** Adaptive behavior with fixed weights in RNN: an overview. *Proceedings* of *International Joint Conference on Neural Networks, 2002,* pages 201 8-2022

[6] Kwan H. K. and Yan J., Second-order recurrent neural network for word sequence learning, *Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing***,** Hong Kong, May 2-4. 2001, pages 405-408.

[7] Tai H.-M.. Wu C.-H., and Jong, T.-L., High-order bidirectional associative memor': *Elecrronics krlers, 25(21):*1424-1425, 12th Oct. 1989.

[8] Jeng Y.-J., Yeh C.-C., and Chiueh T.D., 'Exponential bidirectional associative memories", *Electronics Letters*, 26(11):717-718,24th May1990.

[9] Wang T.. Zhuang X., and Xing X., 'Weight learning of bidirectional associative memories by global minimization", *IEEE Trans. on Neural Networks*, 3(6):1010-1018, Nov. 1992.

[10] Wang Y.F., Cruz J.B. and Mulligan J.H., "Two coding strategies for bidirectional associative memory", *IEEE Trans. Neural Networks*, vol. 1, no. 1, pp 81-92, 1990

[11] Wang Y.F., Cruz J.B. and Mulligan J.H., "Guaranteed recall of all training pairs for bidirectional associative memory", *IEEE Trans. Neural Networks,* vol. 2, no. 6, pp 559-567, 1991

[12] Kang H., "Multilayer associative neural network (MANN's): Storage capacity versus perfect recall*",* *IEEE Trans. Neural Networks*, vol. 5, pp 812-822, 1994

[13] Wang Z., "A bidirectional associative memory based on optimal linear associative memory", *IEEE Trans. Neural Networks,* vol.45, pp 1171-1179, Oct.1996

[14] Hassoun M.H. and Youssef A.M., "A high performance recording algorithm for Hopfield model associative memories", *Opt. Eng*., vol. 27, no. , pp 46-54, 1989

[15] Simpson P.K., "Higher-ordered and interconnected bidirectional associative memories", *IEEE Trans. Syst. Man. Cybern*., vol. 20, no. 3, pp 637-652, 1990

[16] Zhuang X., Huang Y. and Chen S.S., "Better learning for bidirectional associative memory", Neural Networks, vol.6, no.8, pp1131-1146, 1993

[17] Oh H. and Kothari S.C., "Adaptation of the relaxation method for learning bidirectional associative memory*",* *IEEE Trans. Neural Networks*, vol.5, pp 573-583, July 1994

[18] Wang C.C. and Don H.S., "An analysis of high-capacity discrete exponential BAM", *IEEE Trans. Neural Networks*, vol. 6, no. 2, pp 492-496, 1995