Backpropagation with Vector Chaotic Learning Rate

A.M. Numan-Al-Mobin, Mobarakol Islam,
Md. Rihab Rana, Md. Masud Rana
Dept. of Electronics & Communication Engineering,
Khulna University of Engineering & Technology,
Khulna – 9203, Bangladesh

Kaustubh Dhar,Tajul Islam,Md. Rezwan,M. Hossain
Dept. of Electronics & Communication Engineering,
Khulna University of Engineering & Technology,
Khulna – 9203, Bangladesh

Abstract—**In Neural Network (NN) training, local minimum is an integrated problem. In this paper, a modification of standard backpropagation (BP) algorithm, called backpropagation with vector chaotic learning rate (BPVL) is proposed to improve the performance of NNs. BPVL method generates a chaotic time series as Vector form of Mackey Glass and logistic map. A rescaled version of these series is used as learning rate (LR). In BP training the weights of NN become inactive, after arrival of local minima in the training session. Using integrated chaotic learning rate, the weight update accelerated in the local minimum region. BPVL is tested on six real world benchmark classification problems such as breast cancer, diabetes, heart disease, australian credit card, horse and glass. The proposed BPVL outperforms the existing BP and BPCL in terms of generalization ability and also convergence rate.**

Keywords—Neural network; backpropagation; BPCL; BPVL chaos; generalization ability; convergence rate.

I. INTRODUCTION

Gradient based methods are one of the most widely used error minimization methods used to train back propagation networks. The BP training algorithm is a supervised learning method for multi-layered feed forward neural networks [1, 20]. BP have been applied to a wide variety of problems, including pattern recognition, signal processing, image compression, speech recognition etc due to its most appealing features and adaptive nature [2].It is essentially a gradient descent local optimization technique which involves backward error correction of the network weights. Despite the general success of BP in learning the neural networks, several major deficiencies are still needed to be solved [3, 4]. First, the BP algorithm will get trapped in local minima especially for non-linearly separable problems [4]. Having trapped into local minima, BP may lead to failure in finding a global optimal solution [6] second; the convergence rate of BP is still too slow even if learning can be achieved [5]. Furthermore, the convergence behavior of the BP algorithm depends very much on the choices of initial values of connection weights and the parameters in the algorithm such as the learning rate and the momentum. Improving the training efficiency of neural network based algorithm is an active area of research and numerous papers have been proposed in the literature. Early days BP algorithm saw further improvements.

There are many improvement and variations of BP with goals of enlarged speed of convergence, avoidance of local minima and improvement in the network's ability to generalize. BP trains the NNs with constant values of learning rate (LR) and momentum factor. When the LR and momentum factor are made adaptive with the training, the performance of BP is increased [8, 9]. The speed of convergence is accelerated for BP algorithm by using adaptive accuracy of weights, instead of using fixed weight accuracy [10]. Dynamic LR of BP algorithm is optimized by using an efficient method of deriving the first and second derivatives of the objective function with respect to the LR [11]. Another approach that does not require the second order derivative has been proposed to optimize LR. In this case, a set of recursive formula is formed which accelerate the convergence of BP with remarkable savings in running time [12]. Reducing the number of patterns in the active training set effectively increases training efficiency, and accordingly, permits training with a larger pattern set [13]. Many modifications that have been proposed to improve the performance of BP have focused on solving "flat spot" [7] problem to increase the generalization ability. However, their performance is limited due to the error overshooting problem. A novel approach called 2P-MGFPROP has been introduced to overcome the error overshooting problem and hence it speeds up the convergence rate. C. C. Cheung enhanced this approach by dividing the learning process into multiple phases, and different fast learning algorithms are assigned in different phases to improve the convergence rate [15]. All these methods require much computational effort and these do not guarantee good generalization ability for all the cases.

BPCL (Backpropagation with Chaotic Learning Rate) escapes the NN training from premature saturation with chaotic LR. In BPCL to generate chaos Logistic map is use, which is very fast. For this reason we search another learning criteria which is perform much better than BPCL. In this paper, a modified BP algorithm, called 'Backpropagation with Vector Chaotic Learning Rate' (BPVL) is proposed which is a vector form of Mackey glass and Logistic map chaotic time series . BPVL is applied on several benchmark problems including breast cancer, diabetes, heart disease, Australian credit card & horse. For all the problems, BPVL outperforms BP & BPCL in terms of generalization ability and convergence rate.

The rest of the paper is organized as follows. The proposed BPVL is described in section II. Section III includes the experimental studies. The discussion on BPVL is presented in section IV. Section V. concludes the paper.

II. BPVL

BPVL follows the standard BP as well as BPCL. In BPCL the learning rate is chaotic but in BPVL the learning rate is a vector. The Vector is formed by two chaotic time series, one is logistic map [18, 19] and another is Mackey Glass[22,23].

The rescaled LR (RLR) is produced by the following eq^{n} :

$$\mathrm{V(t)=[iC(t) + jM(t)]}$$

$$RLR \ = \mathrm{LR} * \mathrm{abs(V(t))} - \beta$$

β are user specified parameters. β is always kept less than LR close to zero.

C (t) is logistic map time series & M (t) is Mackey Glass time series. However, proposed BPVL is explained as follows.

A feed-forward neural network with one input layer, one hidden layer, and one output layer is shown in Fig. 1. Let, the numbers of input, hidden and output units are *I, J* and *K* respectively. w_{ij} is the network weight that connects input unit *i* and hidden unit *j*, and w_{jk} is the network weight that connects hidden unit *j* and output unit *k*. The number of training examples is *N* and any arbitrary *n*-th training example is $\{x_{n1}, x_{n2}, \ldots\ldots, x_{nI}; y_{n1}, y_{n2}, \ldots\ldots, y_{nK}\}$, where x_n is the input vector and y_n is the target vector. h_{nj} and o_{nk} are the outputs of hidden unit *j* and output unit *k* for the *n*-th training example. Δ represents the difference between the current and new value of the network weights. The consecutive steps of BPVL are given below.

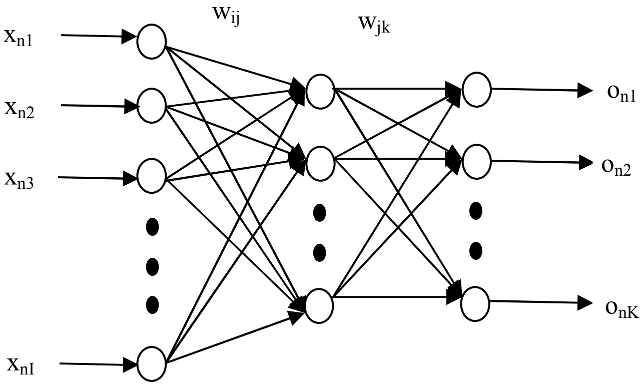

Figure 1. A feed-forward neural network with single hidden layer.

Step 1) Initialize the network weights in an arbitrary interval (-w, +w). Set the iteration number, ITE = 0.

Step 2) Compute the Rescaled LR by using following equation:

$$\mathrm{V(ITE)=abs[iC(ITE) + jM(ITE)]}$$

$$RLR\ \mathrm{(ITE)} = \mathrm{LR} * \mathrm{V(ITE)} - \beta$$

Step 3) Compute h_{nj} and o_{nk} for the n-th training example by using the following:

$$h_{nj} = f\left(\sum_{i=1}^{I} w_{ij}\, x_{ni}\right)$$

$$o_{nk} = f\left(\sum_{j=1}^{J} w_{jk}\, h_{nj}\right)$$

Where, the sigmoid function $f(x) = 1/(1 + e^{-x})$ is used as the activation function. Compute the changes of the weights, Δw_{ij} and Δw_{jk} for the n-th example using the following:

$$\Delta w_{jk} = -RLR\ (ITE)\frac{\partial E_n}{\partial w_{jk}}$$

$$= -RLR\ (ITE)\ \delta_{nk} h_{nj}$$

$$\Delta w_{ij} = -RLR\ (ITE)\delta_{nj} x_{ni}$$

Where,

$$E_n = \tfrac{1}{2}\sum_{k=1}^{K}[o_{nk} - y_{nk}]^2$$

$$\delta_{nk} = (o_{nk} - y_{nk})o_{nk}(1 - o_{nk})$$

$$\delta_{nj} = h_{nj}\left(1 - h_{nj}\right)\sum_{k=1}^{K}\delta_{nk} w_{jk}$$

Update w_{ij} and w_{jk} by adding Δw_{ij} and Δw_{jk} with them correspondingly. When this process is repeated for all the training examples, iteration is completed. ITE is increased by one.

Step 4) Check the termination condition. If the termination condition is fulfilled, the training is stopped and the trained NN is tested; otherwise go to **step 2**.

III. Experimental Studies

A. Characteristics of Benchmark datasets

BPVL is applied on six benchmark classification problems – breast cancer, diabetes, heart disease, Australian credit card, horse, and glass identification. The datasets are collected from the University of California at Irvine (UCI) Repository of the machine learning database [16] and PROBEN1 [17]. Some characteristics of the datasets are listed in TABLE I. For example, cancer is a 2 class problem having total examples of 699 with 9 attributes. Other problems are arranged in a similar fashion. The total examples of a dataset are divided into – training examples, validation examples, and testing examples. The training examples are used to train the NN, validation examples are used to terminate the training process and the trained NN is tested with the testing examples.

TABLE I. Characteristic of Benchmark Datasets.

Datasets	Number of		
	Total Examples	*Input Attributes*	*Output Classes*
Breast Cancer	699	9	2
Diabetes	768	8	2
Heart Disease	920	35	2
Credit Card	690	51	2
Horse	364	58	3
Glass	214	9	6

B. Experimental Process and Comparison

A feed-forward NN with single hidden layer is taken for each problem. The numbers of input and output units of the NN are equal to the number of attributes and number of classes of the dataset respectively. The number of hidden units is taken arbitrarily for several datasets. The weights of the NN are initially randomized in the interval (-1, +1). The training is stopped when the mean square error on the validation set increases in consecutive five iterations. In order to check the generalization performance of trained NN, the 'testing error rate' (TER) is computed. TER is the ration of the number of classified testing examples to the number of testing examples. The experimental results are reported in terms of TER and number of required iterations. Mean and SD indicate the average and standard deviation values of 20 independent trials. BPVL is compared with standard BP & BPCL. To make a fair comparison, the LR of BP is selected in such a way that, this fixed value of learning rate is always an intermediate point in the range [min (RLR) & max (RLR)]

1) Breast Cancer: This problem has 699 examples. The first 350 examples are used as training, 175 examples as validation and last 174 examples for testing. A NN of 9-4-2 (nine input units, four hidden units, and two output units) is taken. The experimental results are listed in TABLE II. When RLR is -0.03 to 0.15, BPVL requires 75.40 iterations to obtain TER of 0.7115(dB), BPCL requires 87.85 iterations to obtain TER of 1.0037(dB) In this range, the LR of BP is considered at 0.06. The TER with BP is 1.9033(dB) and BP requires 152.65 iterations. Here TER is taken in dB 100 times of orginal error Hence BPVL obtains good generalization ability than that of BPCL as well as BP and the number of required iteration with BP is about two times of BPVL and Less than BPCL. Fig. 2 shows the training error of BPVL,BPCL and BP with respect to iteration..

TABLE II. TERS AND NUMBER OF REQUIRED ITERATIONS WITH BP,BPCL AND BPVL FOR CANCER PROBLEM OVER 20 INDEPENDENT RUNS.

Exp. No.	Algorithm	RLR/LR	TER		Iteration	
			Mean(dB)	*SD*	*Mean*	*SD*
01.	BP	.06	1.9033	0.0026	152.65	40.0003
	BPCL	-0.03 to .15	1.0037	0.0029	87.85	23.0636
	BPVL		0.7115	0.0012	75.40	9.6409
02.	BP	0.11	1.2057	0.0026	99.60	18.8981
	BPCL	-0.03 to 0.25	0.7555	0.0017	34.50	5.5812
	BPVL		0.2938	0.0012	32.20	6.4156

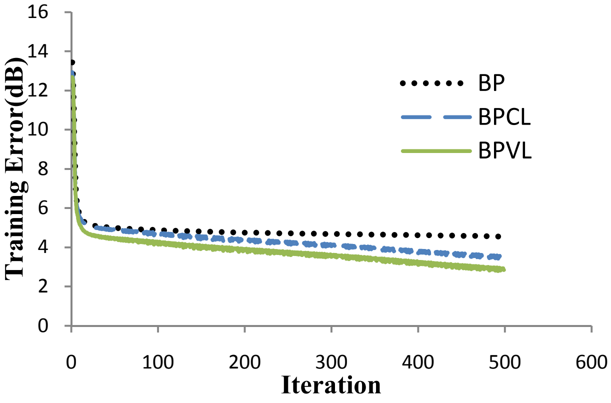

Figure 2. Training error vs. iteration of BP,BPCL and BPVL for cancer problem.

2) Diabetes: This problem has 768 examples .The first 384 examples are used for training, 192 examples for validation and the last 192 examples are used for testing. Here the size of NN is 8-4-2. The results for different LRs are reported in TABLE III. It is shown that BPVL has fast convergence rate than BPCL & BP for all the cases.

TABLE III. TERS AND NUMBER OF REQUIRED ITERATIONS WITH BP, BPCL AND BPVL FOR DIABETES PROBLEM OVER 20 INDEPENDENT RUNS.

Exp. No.	Algorithm	RLR/LR	TER		Iteration	
			Mean(dB)	*SD*	*Mean*	*SD*
01.	BP	.06	13.9058	0.0276	255.95	81.0108
	BPCL	-0.03 to 0.15	13.8650	0.0104	109.90	61.9725
	BPVL		13.8003	0.0066	91.40	15.0176
02.	BP	0.11	13.8970	0.0087	163.65	48.6141
	BPCL	-0.03 to 0.25	13.8686	0.0097	81.70	18.0710
	BPVL		13.7401	0.0070	63.50	6.5459

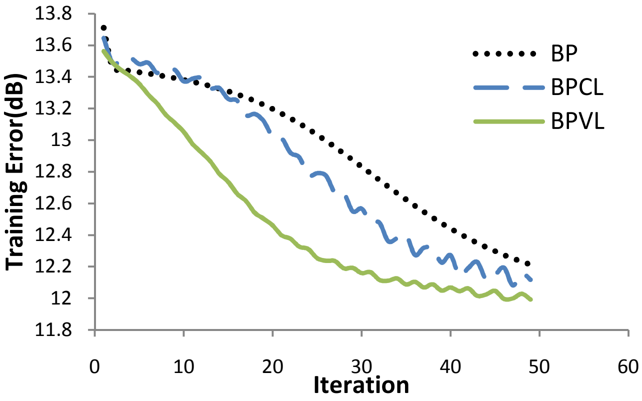

Figure 3. Training error vs. iteration of BP,BPCL and BPVL for diabetes problem.

3) Heart Disease: The size of NN is considered as 35-4-2. This problem has 920 examples. The first 460 examples are used to train the network, 230 examples for validation and the trained network is tested with last 230 examples. The obtained average results are reported in TABLE IV. BPVL outperforms BP & BPCL in all the cases. For example, when RLR is -0.03 to 0.15, TER and iteration of BPVL are 13.0125(dB) and 21.50, while these are 13.1952(dB) and 57.75 for standard BP & 13.0535(dB) and 31.50 for BPCL. The convergence curve of BPVL is also better as shown in Fig. 4.

TABLE IV. TERS AND NUMBER OF REQUIRED ITERATIONS WITH BP, BPCL AND BPVL FOR HEART PROBLEM OVER 20 INDEPENDENT RUNS.

Exp. No.	Algorithm	RLR/LR	TER		Iteration	
			Mean(dB)	*SD*	*Mean*	*SD*
01.	BP	.06	13.1952	0.0085	57.75	26.5191
	BPCL	-0.03 to 0.15	13.0535	0.0092	31.50	17.3882
	BPVL		13.0125	0.0057	21.50	7.8835
02.	BP	0.11	13.1218	0.0096	32.30	16.7186
	BPCL	-0.03 to 0.25	13.0557	0.0086	16.40	7.4659
	BPVL		13.0146	0.0084	13.95	4.6419

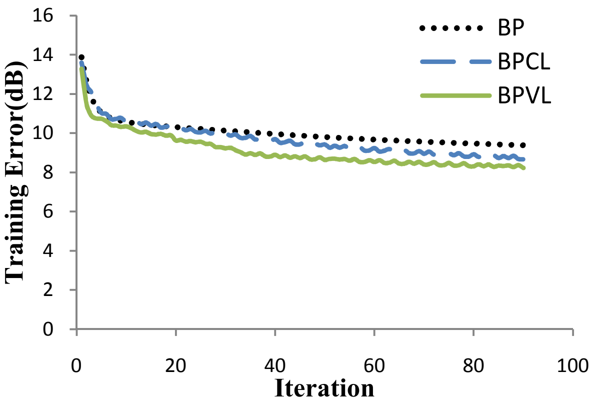

Figure 4. Training error vs. iteration of BP,BPCL and BPVL for heart problem.

4) Australian Credit Card: A 51-4-2 NN is trained with first 345 examples. The training is stopped with 173 validation examples and the NN is tested with last 172 testing examples. TABLE V shows the comparative results between BPCL and BP. When RLR is -0.03 to 0.15, TER and iteration of BPVL are 11.4489(dB) and 22.40, while these are 11.8011(dB) and 41.90 for standard BP 11.5776(dB) & 31.80 for BPCL respectively. These results ensure that the generalization ability of BPVL is better than that of BP & BPCL for credit card problem. Fig. 5 shows the competitive convergence curves of BPVL,BPCL and BP.

TABLE V. TERS AND NUMBER OF REQUIRED ITERATIONS WITH BP, BPCL AND BPVL FOR CARD PROBLEM OVER 20 INDEPENDENT RUNS.

Exp. No.	Algorithm	RLR/LR	TER		Iteration	
			Mean(dB)	*SD*	*Mean*	*SD*
01.	BP	.06	11.8011	0.0112	41.90	8.1234
	BPCL	-0.03 to 0.15	11.5776	0.0089	31.80	4.8846
	BPVL		11.4489	0.0107	22.40	2.9899
02.	BP	0.11	11.8013	0.0096	23.95	6.9700
	BPCL	-0.03 to 0.25	11.4737	0.0072	19.10	4.1901
	BPVL		11.1461	0.0810	14.65	3.7319

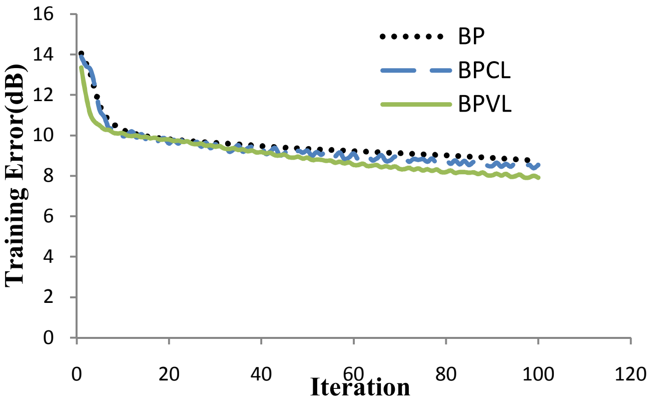

Figure 5. Training error vs. iteration of BP, BPCL and BPVL for card problem.

5) Horse: The horse dataset has total 364 examples. The first 182 examples are used for training, 91 examples for validation and last 91 examples for testing. A 58-7-3 NN is trained. The numerical results are reported in TABLE VI and the error curves are shown in Fig. 6. When RLR is -0.02 to 0.15, TER and the number of required iterations of BPVL are 11.5776(dB) and 30.40 while these are 14.6419(dB) and 47.05 for standard BP,. 14.5163(dB) and 38.5 for BPCL respectively.

TABLE VI. TERS AND NUMBER OF REQUIRED ITERATIONS WITH BP, BPCL AND BPVL FOR HORSE PROBLEM OVER 20 INDEPENDENT RUNS.

Exp. No.	Algorithm	RLR/LR	TER		Iteration	
			Mean(dB)	*SD*	*Mean*	*SD*
01.	BP	.06	14.6419	0.0185	47.05	13.3958
	BPCL	-0.02 to 0.15	14.5163	0.0271	38.55	13.9373
	BPVL		14.4576	0.0135	30.40	7.0092
02.	BP	0.11	14.6090	0.0301	36.20	12.3150
	BPCL	-0.02 to 0.25	14.3965	0.0201	24.90	6.7742
	BPVL		14.2911	0.0233	23.55	12.214

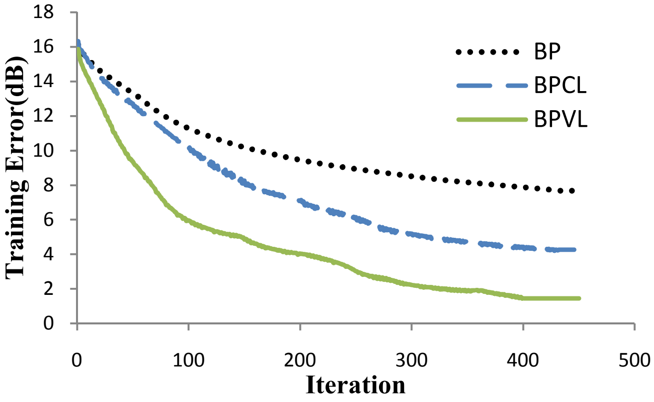

Figure 6. Training error vs. iteration of BP, BPCL and BPVL for horse problem.

6) Glass: The glass dataset has total 214 examples. The first 107 examples are used for training, 54 examples for validation and last 53 examples for testing. A 9-6-6 NN is trained. The numerical results are reported in TABLE VII and the error curves are shown in Fig. 7. When RLR is -0.02 to 0.15, TER and iteration of BPVL are 15.9295(dB) and 131.60, while these are 16.0239(dB) and 186.55 and 16.2407(dB) and 410.50 for BPCL and standard BP respectively. Here the numbers of required iterations for all examples with BP are about four times than that of with BPVL. Fig. 7 shows that BPCL has better convergence rate than that of BP.

TABLE VII. TERS AND NUMBER OF REQUIRED ITERATIONS WITH BP,BPCL AND BPVL FOR GLASS PROBLEM OVER 20 INDEPENDENT RUNS.

Exp. No.	Algorithm	RLR /LR	TER		Iteration	
			Mean(dB)	SD	Mean	SD
01.	BP	0.06	16.2407	0.1307	410.50	171.98
	BPCL	-0.02 to 0.15	16.0239	0.1367	186.55	79.09
	BPVL		15.9295	0.0680	131.60	31.20
02.	BP	0.11	16.1109	0.0719	360.00	175.12
	BPCL	-0.02 to 0.25	15.4986	0.0670	91.12	21.60
	BPVL		15.4370	0.0256	80.51	15.78

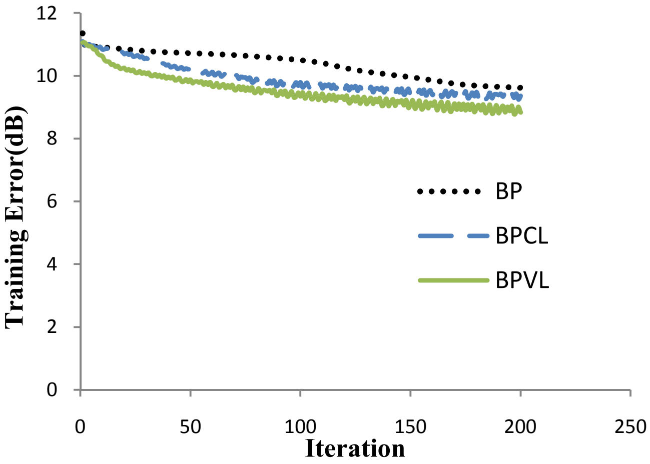

Figure 7. Training error vs. iteration of BP, BPCL and BPVL for glass problem.

IV. Discussion

The difference between BPCL and BPVL is that BPCL makes the LR chaotic during the training by logistic map while BPVL trains the network by a Vector learning rate, which is a vector form of two chaotic time series, Mackey glass and Logistic map. Logistic map is the fastest chaotic time series; that is why consecutive learning rate difference is high in BPCL. To optimize this problem vector learning rate is used in BPVL. The training error curve of BPVL is improved and sometimes it is marginally improved than that of BPCL & BP. However, BPCL is proposed to achieve a good generalization ability, not to improve the convergence, although BPVL shows fastest convergence too.

The NN training is terminated based on validation examples. When the mean square error on the validation set starts to increase, BPVL stops the training. There is a possibility to fall the validation error into local minima and so the mean square error is checked in five consecutive iterations.

V. Conclusion

In this paper, a supervised training algorithm, called BPVL is proposed. BPVL works on the learning parameter of training. Standard BP trains NNs with a constant value of LR. On the other hand, biological systems such as human brain involve chaos. To fill up this gap, BPVL trains NNs with LR which is a vector chaotic time series. The chaotic series is generated by using the complex vector form of the Mackey Glass and logistic map and a rescaled version of this is intentionally incorporated with the training. Due to the nonlinear error surface of real world problem, BP training often falls into premature saturation that leads the training to a non updating zone. Since most of the time, the LR is positive and sometimes it is negative, BPVL resolves this problem.

BPVL is applied on six benchmark classification problems to observe the training performance. BPVL is capable to train NNs with better generalization ability and also faster convergence rate than that of standard BP and BPCL.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable suggestion.

References

[1] M. K. S. Alsmadi, K. B. Omar, and S. A. Noah, "Back Propagation Algorithm: The Best Algorithm Among theMulti-layer Perceptron Algorithm", International Journal of Computer Science and Network Security, vol. 9, no. 4, April 2009.

[2] X. Zhou, J. Zheng, and K. Mao, "Application of Modified Backpropagation Algorithm to the Prediction of the Chloride Ion Concentration in Cracked Concrete", In the Proc. of 3rd International Conference on Natural Computation, pp. 257-261, 2007.

[3] S. Haykin, "Neural Networks, A Comprehensive Foundation", New York 10022: IEEE Society Press, Macmillan College Publishing, *1994.*

[4] S. E. Fahlman, "An empirical study of learning speed in back propagation networks*,"* tech. rep., CMU-CS-88-162*,* Carnegie Mellon University, Pittsburgh, PA., 1988.

[5] M. A. Otair and W. A. Salameh, *"*Speeding Up Back-Propagation Neural Networks", Proceedings of the Informing Science and IT Education Joint Conference, 2005.

[6] M. Gori and A. Tesi, "On the problem of local minima in backpropagation*,"* IEEE Trans. Pattern Anal. Machine Intell., vol. 14, pp. 76–86, 1992.

[7] J. E. Vitela and J. Reifman, "Premature saturation in backpropagation networks: Mechanism and necessary conditions," Neural Netw., vol. 10, no. 4, pp. 721–735, 1997.

[8] C. C. Yu, and B. D. Liu, "A Backpropagation Algorithm With Adaptive LR And Momentum Coefficient", In the Proc. of International joint Conference on neural network, pp. 1218-1223, 2002.

[9] S. Ng, C. Cheung, and S. Leung, "Magnified Gradient Function With Deterministic Weight Modification in Adaptive Learning", IEEE trans. neural netw, vol. 15, no. 6, 2004.

[10] C. Gonzalo, Q. Tian, Y. Fainman, and S. H. Lee, "Fast Convergence of the Backpropagation Learning Algorithm by Using Adaptive Accuracy of Weights", Proceedings of the 35th Midwest Symposium on Circuits and systems, vol. 2, pp. 1221-1224, 1992.

[11] X. H. Yu, "Dynamic LR Optimization of the Backpropagation Algorithm*"*, IEEE Transactions on Neural Networks*,* vol. 6, no. 3, May 1995.

[12] X. H. Yu and G. A. Chen, "Efficient estimation of dynamically optimal LR and momentum for backpropagation learning", In the Proc. of IEEE International Conference on Neural network, vol. 1, pp. 385-388, 1995.

[13] E. M. Strand and W. T. Jones, "An Active Pattern Set Strategy for Enhancing Generalization While Improving Backpropagation Training Efficiency", In Proc. of International Joint Conference on neural network, vol. 1, pp. 830-834, 1992.

[14] C. C. Cheung and S. C. Ng, "Backpropagation with Two-Phase Magnified Gradient Function", Proceedings of IEEE World Congress on Computational Intelligence (WCCI), Hong Kong, 2008.

[15] Chi-Chung Cheung, "The Multi-Phase Method in Fast Learning Algorithm", Proceeding of international join conference on neural networks, Atlanta Georgia, USA, June 14-19, 2009.

[16] A. Asuncion and D. Newman, ''UCI Machine Learning Repository*",* Schl. Inf. Comput. Sci., Univ. California, Irvine, CA, 2007.

[17] L. L. Prechelt, "PROBEN1-A set of neural network benchmark problems and benchmarking rules", Technical Report 21/94, Faculty of Informatics, University of Karlsruhe, 1994.

[18] Junshan Gao, Baiyu Sun and Jiaxiang Yang, "The Construction of the Cubic Logistic Chaotic Function Family", In Proc. of Sixth World Congress on Intelligent Control and Automation, 2006.

[19] Olivares, E. I. Vazquez-Medina, R. Cruz-Irisson, and M. Del-Rio-Correa, "Numerical calculation of the lyapunov exponent for the logistic MAP", 12th International Conference on Mathematical Methods in Electromagnetic Theory, 2008.

[20] Rumelhart, D.E., G.E. Hinton and R.J. Williams, “Learning Internal Representations by Error Propagation”, Parallel Distributed Processing,MIT ,pp: 318-362, 1986.

[21] Mobarakol Islam, Md. Rihab Rana, S.U. Ahmed, Md Shahjahan, “Training Neural Network with Chaotic Learning Rate”, In the proc. of International Conference on Emerging Trends in Electrical and Computer Technology, March 23-24, 2011.

[22] M. Farzad, H. Tahersima and H. Khaloozadeh, “Predicting the Mackey Glass Chaotic Time Series Using Genetic Algorithm”, Proceeding of International Joint Conference, SICE-ICASE, pp. 5460 – 5463, 2006.

[23] W.C Mead, R.D Jones, Y.C. Lee, C.W. Barnes, G.W. Flake, L.A. Lee and M.K. O'Rourke, “Using CNLS-net to predict the Mackey-Glass chaotic time series” In the proc. of International Joint Conference on Neural Networks(IJCNN) , vol.2, pp. 485 – 490, 1991.

[24] Gupta, S., Doshi, V., Jain, A., & Iyer, S. (2010). Iris Recognition System using Biometric Template Matching Technology. *International Journal of Advanced Computer Science and Applications - IJACSA*, *1*(2), 24-28. doi: 10.5120/61-161.

A. M. Numan-Al-Mobin received the B.Sc. degree in Electronics and Communication Engineering from Khulna University of Engineering & Technology, Bangladesh, 2008. He was a Rollout Junior Engineer, July 2008 to October 2008, in Orascom Telecom Bangladesh Limited (Banglalink). He is currently working as a lecturer in the same department of the university. His current research interests include electromagnetic, antennas and propagation, wireless communication and signal processing.

Mobarakol Islam received the B.Sc. degree in Electronics & Communication Engineering from Khulna University of Engineering & Technology,Bangladesh, 2011. He is presently working in the Samsung Bangladesh Research & Development Centre as Software Engineer-1. His research interest include machine learning, pattern recognition, neural networks, image processing, telecommunication and feature selection.

Md. Rihab Rana received B.Sc. degree in Electronics & Communication Engineering from Khulna University of Engineering & Technology,Bangladesh, 2011. His research interest include machine learning,neural networks, feature selection and telecommunication.

Md. Masud Rana is currently the Assistant Professor at Department of Electronics & Communication Engineering, Khunla University of Engineering & Technology, Bangladesh. He received the B.Sc in Electronics & Communication Engineering and M.Eng. in Electronics & Radio Engineering degrees from Khunla University of Engineering & Technology and Kyung Hee University, Korea, respectively. His current research interests include broadband digital communications systems, channel estimation and tracking, interference prediction and cancellation, equalization, digital signal processing in communications, OFDM, cooperative communications as well as single and multicarrier transmission. He is a member of the Institute of Electronics Engineers of Korea (IEEK), Institute of Engineers Bangladesh (IEB), live member of Bangladesh Electronics Society (BES), and student member of IEEE.

Kaustubh Ratan Dhar received B.Sc. degree in Electronics & Communication Engineering from Khulna University of Engineering & Technology,Bangladesh in 2011. His reaearch field interests include Neural Networks, Telecommunication,Numerical computation of electromagnetic fields, integrated optical devices and waveguide materials

Md. Tajul Islam received B. Sc. degree in Electronics & Communication Engineering from Khulna University of Engineering & Technology,Bangladesh, 2011. His reaearch field is Neural Networks,Telecommunication and VLSI chip design.

Md. Rezwanul Haque received B. Sc. degree in Electronics & Communication Engineering from Khulna University of Engineering & Technology,Bangladesh, 2011. His reaearch field is Neural Networks,Telecommunication and VLSI chip design.

Md. Moswer Hossain received B. Sc. degree in Electronics & Communication Engineering from Khulna University of Engineering & Technology,Bangladesh, 2011. His reaearch field is Neural Networks, feature selection and Telecommunication .