A Novel Variable Step Size LMS Algorithm Based On Neural Network

Anyang Zhang   Ningsheng Gong

College of Information Science and Engineering, Nanjing University of Technology, Nanjing 210009, P. R. China

Abstract

This paper presents a new variable step size LMS algorithm based on neural network (BP-LMS). A non-linear relationship amongst the input vectors, deviation errors and the learning steps is constructed by BP model, which is employed to determine the learning steps during adaptive processing. The proposed algorithm also takes the prior knowledge into the LMS algorithm. Simulation experiments suggest that BP-LMS algorithm is capable of decreasing the time of convergent progress rapidly and satisfactory performance is attainable even with the presence of high level of noise.

Keywords: Adapt filtering, Variable step size LMS algorithm, BP neural network, Prior knowledge

1. Introduction

Adaptive signal processing has been widely applied to system identification, sonar technology, image processing and related fields[1]. The filter parameters can be iteratively adjusted optimally based on adaptive filter theory. A simple but effective algorithm named Least-Mean-Square algorithm (LMS) was developed by Widrow and Hoff in 1960. Determining the learning step μ is an essential problem in LMS algorithm. If μ is selected too small the convergent speed of the algorithm will be very slow, or severe mis-adjustment(that is Mean Square Error, MSE) can easily cause instability if μ is too large[2]-[3]. Although Widrow presented the range of learning step μ ($0 < \mu < 1/\lambda_{max}$, λ_{max} is the maximal eigenvalue of input signals) which ensures the algorithm to be convergent, the choice of optimal learning step has not been properly addressed. A lot of work has been down to compromise the convergent speed and stability. Most of these approaches are, however, simply presenting a sampling function that relates the learning steps and the mis-adjustments[4]-[5]. The present study proposes a new variable step size LMS algorithm, whose learning steps are controlled by a BP neural network (BP-LMS). We use a BP neural network to construct a mode that amongst the input vectors, deviation errors and the learning steps, in which the prior knowledge is also taken into account. Moreover, experimental results demonstrate that BP-LMS algorithm is capable of making the filter convergent with an enhanced speed while the filter is still operating in a stable state.

This paper is organized as follows: Section 2 and Section 3 are used to briefly introduce the principle of LMS adaptive filter and the BP neural network, respectively. Section 4 presents BP-LMS and its realization. And Experimental results are reported in section 5. Finally, some conclusions are drawn in section 6.

2. LMS Adaptive Filter

The realization of adaptive filter is shown in Fig. 1.

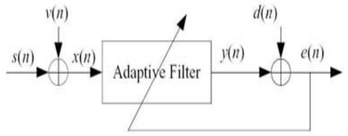


Fig. 1: Adaptive filter principle.

Here s(n), v(n), d(n) and e(n) are signal-of-interest, noise, desired signal and error respectively. LMS algorithm is described by the following equations:

$$e(n) = d(n) - X^T(n) * W(n)$$

$$W(n+1) = W(n) + 2\mu e(n) X^T(n)$$

where μ is learning step, X(n) is the input vector at sampling time n, W(n) is the coefficient vector of the adaptive filter, d(n) is the expected output value, e(n) is the deviation error, The dimension of W(n) is the length of the adaptive filter.

Convergent speed and stable mis-adjustment are two important factors for evaluating the performance of an adaptive filter. In generally, decreasing the learning step μ can reduce the stable mis-adjustment of the filter, but slows down the convergent progress. When increasing μ, the convergent speed is enhanced, but the mis-adjustment of the filter output is increased with undesirable effects.

Many researchers presented a lot of algorithms to solve the contradiction between the convergent speed and stable mis-adjustment. However, litter attention has been paid to the non-linear relationship amongst the input vectors, deviation errors and the learning steps in these algorithms. On the other hand, the way to find the optimal learning steps used before in these algorithms are insignificant in the later stage of adaptive processing, in other words, the prior knowledge wasn't taken into account.

In this paper, we want to take measures to look for optimal learning steps to make the stable mis-adjustment be minimal with suitable convergent speed. We assume to present a new LMS algorithm, in which the learning steps are affected by the input vectors and deviation errors, while the experience in looking for the optimal learning steps used before can contribute to find the optimal learning steps in later stage of adaptation process.

ANN(Artificial neural network) is applied in this paper since much intelligence such as self-learning and strong ability in non-linear solutions can be extracted and further utilized. Using neural network to determine the learning step in adaptive filter is a good way which can overcome the contradiction between convergent speed and stable mis-adjustment.

3. BP neural network

In recent years, people pay much attention to ANN because of its distinctive information handling and its unique calculation ability. ANN uses a flock of neural cells to form a net, which can approximate to any complex non-linear system. ANN has a lot of significant features such as self-learning, self-organization, self-adapting, fault-tolerant and so on[6]. At present, ANN has provided good results in many applications, especially MFNN (Multilayer Feed-forward Neural Network), which has been an effective tool in solving engineering problems[7].

Since hidden neural cells have been introduced into ANN, the ANN has much more capability of classification and recollection. The feed-forward, back-propagation architecture, developed in the early 1970's, is the most popular, effective, and easy to learn model for complex, multi-layered networks. It is used in many different types of applications and its greatest strength is in non-linear solutions to ill-defined problems. A typical back-propagation(BP) network has an input layer, an output layer, and at least one hidden layer, Each layer is connected to the succeeding layer, as shown in Fig.2.

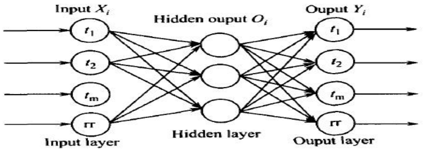


Fig.2: An Example of three-layer BP Network.

The principle of BP network is that: the input value X_i act on the output node through the nodes in the hidden layer, which generates the output value Y_k by nonlinear transformation. Each sample for network training includes: the input value X, the desired output t, and the deviation between output value Y and desired output t. The deviation can be declined in gradient trend by adjusting the connection weights W_{ij} between the node i in the input layer and the node j in the hidden layer, the connection weights T_{jk} between the node j in the hidden layer and the node k in the output layer, and the threshold. By repeating the studying and the training, the BP model parameters(connection weights and threshold) according to the min-allowed- error will be gotten, and the training will be completed[9]-[10].

After the BP model trained, the BP model parameters imply certain non-linear relationship between the input signal and the output signal. Corresponding response can be attained by putting input data into the trained BP network.

4. BP-LMS

In most variable step LMS algorithms, the learning steps are determined by the deviation errors through certain sampling function, whose performance are not very good. In BP-LMS, a relationship among the input vectors, deviation errors and the learning steps is built, it means that the learning steps are affected by the current input signals and the deviation errors, and methods to find the optimal learning steps used before are also valuable in later filtering process.

BP neural network used here to make the BP-LMS come true since its greatest strength in non-linear solutions[7]-[8]. Fig.3 shows the principle of the BP-LMS used in adaptive filter, where s(n) denotes the signal-of-interest at the sampling time n

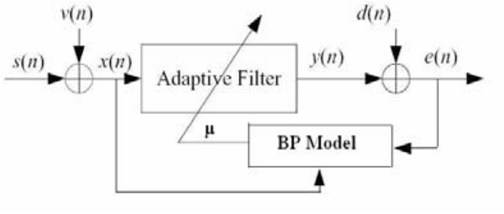


Fig.3: BP-LMS used in adaptive filter.

and v(n) is noise, which is uncorrelated with s(n). D(n) and e(n) denote desired signal and error respectively. We can see from Fig.3 that the e(n) and x(n) at sampling time n are also employed as the input vectors of the BP model. The learning step μ is the output of the BP network. Thus the essence of the BP-LMS algorithm is to construct a BP model to look for the optimal steps according to the input vectors and the deviation errors, which means we have to look for a suitable training sample for the BP network in the BP-LMS during the training process, by which the relationship amongst the input vectors, deviation errors and learning steps and the prior knowledge can be integrated into the BP-LMS algorithm.

Greedy algorithm is employed here to search training samples for the BP network. We test every step in the stability interval to look for the learning steps which can make the next deviation errors become minimal. The input vectors, deviation errors and the optimal learning steps are recorded as the BP-LMS' training sample. After the training sample has been prepared, it is not difficulty to construct and train a BP model. Detailed principle and algorithm of BP model can be seen in Ref. [9]-10].

After the BP model trained, the parameters of the BP model not only represent the non-linear relationship amongst the input vectors, deviation errors, and the optimal learning steps, but also manifest the effect of the prior knowledge. Thus the BP-LMS algorithm updates the filter parameters based on the following equations:

$$e(n) = d(n) - X^T(n) * W(n)$$
$$W(n+1) = W(n) + \mu e(n) X^T(n)$$
$$\mu = \begin{cases} \mu_{\max}, \beta > \mu_{\max} \\ \mu_{\min}, \beta < \mu_{\min} \\ \beta, else \end{cases}$$

In the equations listed above, β is the output of the BP model according to the input vector x(n) and deviation error e(n) at the sampling time n. To ensure the BP-LMS' convergence, β must be less than $1/\lambda_{\max}$ ($\lambda_{\max}$ is the maximal eigenvalue of input signals), and being positive. Instead of zero, we use $\mu_{\min}$(a litter decimal fraction) to ensure that μ is still influenced by the variation of error and the input signal while β is very close to zero so as to improves the convergence speed of the algorithm. In practical applications, the $\mu_{\max}$ and $\mu_{\min}$ can be attained by experiments[5].

Once the BP model's training work has been finished, the optimal learning steps can simple be available during the adaptive process. We only have to make the current input vector and deviation error as the input data of the BP model, then the current optimal learning step will be outputted by the trained BP model.

Based on neural network, we have successfully used the prior knowledge into LMS algorithm with the filter being guaranteed convergent more quickly. It also, to some extent, shows the BP-LMS has the ability in intelligence controlling. Simulations are performed in next section to show the performance of the BP-LMS algorithm.

5. Simulation Results

To testify the rationality and efficacy of the BP-LMS algorithm, simulations are performed and three other LMS algorithms are employed to compare the performance with the BP-LMS algorithm. One is standard LMS algorithm, whose learning step is a small positive invariable constant. Another is NLMS(Normalized Least Mean Square) algorithm, in which the learning step can be obtained by the following equation:

$$\mu = \frac{\eta}{\delta + \| X(n) \|^2}$$

Where η is an adaptation constant, δ is a positive number, which ensures the NLMS convergent if the input vector norm is small and X(n) is regarded as the input signal at the sampling time n. The last is VSS-LMS(Variable Step Size Least Mean Square) algorithm[11], whose learning step μ(n+1) is determined by the following equations:

$$e(n) = d(n) - X^T(n) * W(n)$$
$$W(n+1) = W(n) + \mu(n+1) e(n) X^T(n)$$
$$\mu(n) = \begin{cases} \mu_{\max}, \mu(n) > \mu_{\max} \\ \mu_{\min}, \mu(n) < \mu_{\min} \\ \mu(n), else \end{cases}$$
$$\mu(n+1) = \alpha \mu(n) + \gamma e^2(n)$$

Where α and γ are two parameters of the VSS-LMS algorithm, e(n) and μ(n) are deviation error and learning step at the sampling time n. $\mu_{\min}$ and $\mu_{\max}$ play the same roles what we talk about in BP-LMS algorithm.

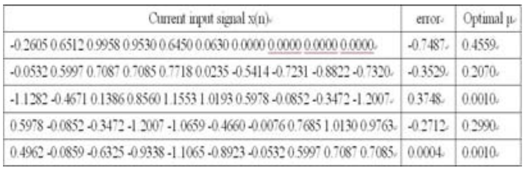
Simulation I: comparison amongst the four algorithms when the input signal is standard sinusoidal signal:

- The order of the adaptive filter is set to L=10;
- The initial weight W(n) of the adaptive filter is defined as zero.
- The signal-of-interest s(n) is standard sinusoidal signal.
- v(n) is white Gaussian sequence, which is uncorrelated with s(n).
- The invariable learning step μ in standard LMS algorithm is set to 0.005, and two

parameters η and δ of the NLMS adaptive filter are defined as 0.15 and 2 respectively. Parameters α and γ in VSS-LMS algorithm are set to 0.97 and 0.0008 respectively while μ_{max} and μ_{min} in this algorithm are found at 0.008 and 0.0001. And the initial learning step μ(0) is set to 0.003.

- Statistical mean times and samples are 20 and 1000, respectively.

In BP-LMS adaptive filter, firstly we have to use greedy algorithm to search the training sample for the BP model, part of the training data are listed in Fig.4.



Current input signal x(n)	error	Optimal μ
-0.2605 0.6512 0.9958 0.9530 0.6450 0.0630 0.0000 0.0000 0.0000 0.0000	-0.7487	0.4559
-0.0532 0.5997 0.7087 0.7085 0.7718 0.0235 -0.5414 -0.7231 -0.8822 -0.7320	-0.3529	0.2070
-1.1282 -0.4671 0.1386 0.8560 1.1553 1.0193 0.5978 -0.0852 -0.3472 -1.2007	0.3748	0.0010
0.5978 -0.0852 -0.3472 -1.2007 -1.0659 -0.4660 -0.0076 0.7685 1.0130 0.9763	-0.2712	0.2990
0.4962 -0.0859 -0.6325 -0.9338 -1.1065 -0.8923 -0.0532 0.5997 0.7087 0.7085	0.0004	0.0010

Fig.4: part sample data for the BP model to train in simulation I.

A BP model then is constructed, with an 11-component input vector(10 for the input signal of the adapt filter and one for the error produced under the assumed conditions), 25 hidden units and scale output for the BP model to producing the optimal learning step μ. After several experiments, the best μ_{max} and μ_{min} are found at 0.2 and 0.005, respectively.

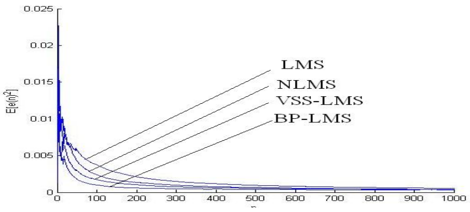


Fig. 5: Performance of the four algorithms in low noise in simulation I.

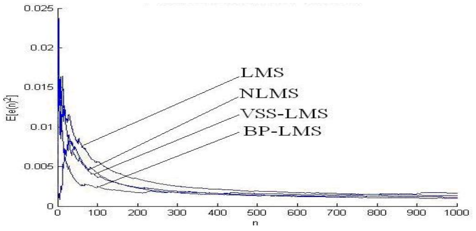


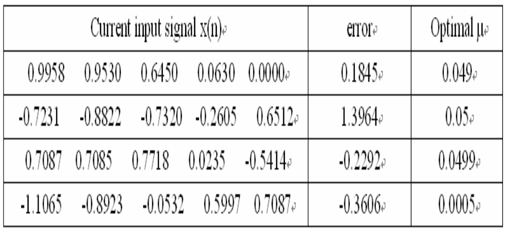
Fig. 6: Performance of the four algorithms in high noise in simulation I.

Fig. 5 presents the performance of four methods subjected to low of noise. In this simulation, the noise added to the signal-of-interest is a zero-mean, independent Gaussian random sequence with 0.04 variance. From Fig. 5, it can be seen that the BP-LMS algorithm is capable of making the adaptive filter convergent at a much enhanced speed, about less than 100 times, that is much more rapidly than the other three algorithms. To further evaluate the performance of the algorithm, from Fig. 6, we can see that filter adjusted by the BP-LMS algorithm can be converges quickly even when the noise power was increased to 0.09. Only about 150 iterations are required whereas the VSS-LMS, NLMS and LMS algorithms, respectively, took more than 250, 300 and 400 iterations to achieve the similar level of error reduction.

Simulation II: comparison amongst the four algorithms when the input signal is white Gaussian sequence:

- The order of the adaptive filter is set to L = 5.
- The weight of the adaptive filter is defined as $W(n) = [0.25, 0.75, 1, 0.75, 0.25]^T$.
- The signal-of-interest s(n) is zero-mean white Gaussian random with variance $\sigma^2 = 1$.
- v(n) is white Gaussian sequence, which is uncorrelated with s(n).
- The invariable learning step μ in LMS algorithm is set to 0.005, and two parameters η and δ of the NLMS adaptive filter are defied as 0.15 and 2 respectively. Parameters α and γ in VSS-LMS algorithm are set to 0.97 and 0.0008 respectively. μ_{max} and μ_{min} in this algorithm are found at 0.05 and 0.0005. And the initial learning step μ(0) is set to 0.005.
- Statistical mean times and samples are 200 and 1000, respectively.

Like in simulation I, greedy algorithm is employed firstly to look for the training sample for the BP model, part of the training data are listed in Fig. 7.



Current input signal x(n)	error	Optimal μ
0.9958 0.9530 0.6450 0.0630 0.0000	0.1845	0.049
-0.7231 -0.8822 -0.7320 -0.2605 0.6512	1.3964	0.05
0.7087 0.7085 0.7718 0.0235 -0.5414	-0.2292	0.0499
-1.1065 -0.8923 -0.0532 0.5997 0.7087	-0.3606	0.0005

Fig.7 part sample data for the BP model to train in simulation II.

A BP model then is built, with an 6-component input vector(5 for the input signal of the adapt filter and one for the error produced under the assumed conditions), 15 hidden units and scale output for the BP model to producing the optimal learning step μ. After several experiments, the best μ_{max} and μ_{min} are found at 0.05 and 0.0005, respectively. Fig. 8 presents the performance of four algorithms subjected to low of noise, which was a zero-mean, independent Gaussian random sequence with 0.01 variance. In Fig.9, the power of the noise added to signal-of-interest was increased to 0.04. We can obtain from the two figures that, despite of the power of the noise, the BP-LMS algorithm has a faster convergence speed than three other algorithms at the same stable mis-adjustment.

From the two simulations experimental results, they all illustrate the efficacy of the BP-LMS algorithm, as well as its rationality.

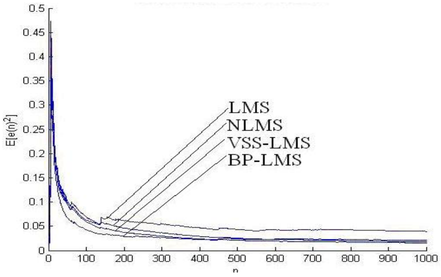


Fig.8 Performance of the four algorithms in low noise in simulation II.

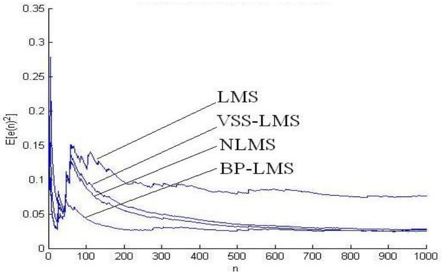


Fig.9: Performance of the four algorithms in high noise in simulation II.

6. Conclusions

In this paper, a BP-LMS algorithm employing a new adaptive step size control technique was proposed. The new algorithm, whose learning step is controlled via a BP neural network, possesses desirable features that are significant for adaptive filtering applications. Simulations experimental results have demonstrated the BP-LMS algorithm's advantageous features and its ability to maintain convergence rate improvements with small mis-adjustment under different environmental conditions.

References

[1] H.K. Simon, editor. *Adaptive filter theory, Fourth Edition*, Prentice Hall, 2001.

[2] R. Kwong and E.W. Johnston, A Variable Step Size LMS Algorithm, *IEEE Trans. On Signal Process.*, 40(7): 1633-1642, 1992.

[3] T. Aboulnasr and K. Mayyas, A robust variable step size LMS-type algorithm: analysis and simulations, *IEEE Signal Process.*, 45: 631–639, March 1997.

[4] Y. Gao and SH.L. Xie, A Variable Step Size LMS Adaptive Filtering Algorithm and Its Analysis, *Acta. Electronica Sinica*, 29: 1094-1097, 2001.

[5] J.F. Qin and J.ZH. Ou.Yang, A new variable step size adaptive filtering algorithm, *Data Collecting and Processing*, 12(3): 171-194, 1997.

[6] R. Rojas, editor. *Neural Networks--A Systematic Introduction*. Springer, 1996.

[7] K.C. Tan and H.J. Tang, New Dynamical Optimal Learning for Linear Multilayer FNN, *IEEE Transactions on Neural Networks (S1045-9227)*, Accepted for future publication, 99: 1-7, 2004.

[8] Y.C. Liang and Z.S. Wang. A preprocessing method on dense type data using artificial neural network BP algorithm. *Acta Sci Natural Univ Jilinensis*, 3: 19–22, 1995.

[9] P.F. Yan, ediror, *Artificial neural network and Analogy Computation*, Tsinghua University Press, 2000.

[10] J. Gao, editor. *The Principle of Artificial Neural Network and Analogy Example,* Mechanics Industry Press, 2003.

[11] Q.P. Wang, D.L. Tarn and Y.C. Wang, Event-based Intelligent Control System of Carbide Electric Arc Furnace(CEAF), *Proceedings of the 3rd World Congress on Intelligent Control and Automation [C]*. Hefei, China:Press of University of Sience and Technology of China,bpp. 471～476, 2000.