(accepted and in press,confidential and non-distributed)

Fully Complex Extreme Learning Machine

Ming-Bin Li, Guang-Bin Huang *, P. Saratchandran, and N. Sundararajan

School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798

Abstract

Recently, a new learning algorithm for the feedforward neural network named the extreme learning machine (ELM) has been proposed by Huang, et al, which can give better performance than traditional tuning-based learning methods for feedforward neural networks in terms of generalization and learning speed. In this paper, we first extend the ELM algorithm from the real domain to the complex domain, and then apply the fully complex extreme learning machine (C-ELM) for nonlinear channel equalization applications. The simulation results show that the ELM equalizer significantly outperforms other neural network equalizers such as the complex minimal resource allocation network (CMRAN), complex radial basis function (CRBF) network and complex backpropagation (CBP) equalizers. C-ELM achieves much lower symbol error rate (SER) and has faster learning speed.

Key words: Feedforward neural networks, complex QAM equalization, complex extreme learning machine, complex activation function, CMRAN, CRBF, CBP.

1 Introduction

In high speed digital communication systems, equalizers are used very often at receivers to recover the original symbols from the received signals. Real-valued neural network models such as feedforward neural networks, radial basis function (RBF) networks and recurrent neural networks have been successfully used for solving equalization problems as neural networks are well suited for non-linear classification problems [3]. Complex-valued neural networks have attracted considerable attention in channel equalization applications in the

* Corresponding author
Email address: egbhuang@ntu.edu.sg (Guang-Bin Huang).

past decade. Cha and Kassam [1] have proposed a Complex-valued Radial Basis Function(CRBF) network which adopts the stochastic gradient learning algorithm to adjust parameters. Compared with previously existing equalizers, the CRBF equalizer is superior in terms of symbol error rate (SER). Jianping et al [8] have developed a Complex-valued Minimal Resource Allocation Network (CMRAN) equalizer. Applying the growing and pruning criterion, the CMRAN equalizer realizes a more compact structure and obtains better performance than CRBF and many other equalizers. However, it should be noted that although the inputs and the centers of CRBF and CMRAN are complex-valued, the basis functions still remain real-valued. In fact, as pointed out by Kim and Adali [10], split-complex activation (basis) functions consisting of two real-valued activation functions, one processing the real part and the other processing the imaginary part, have been traditionally employed in these complex-valued neural networks. Kim and Adali [10,9] have proposed an important complex neural network model - a fully complex multilayer perceptron (MLP) which uses true complex-valued activation function. It has been rigorously proved [10] that with very mild condition on the complex activation functions the fully complex MLPs can universally approximate any continuous complex mappings. The corresponding fully complex backpropagation (CBP) learning algorithm with fully complex activation function has also been successfully used in communication applications [9].

Recently, a new learning algorithm for Single-hidden-Layer Feedforward Neural network (SLFN) named the extreme learning machine (ELM) has been proposed by Huang, et al[7,6]. Unlike traditional approaches (such as BP algorithms) which may face difficulties in manually tuning control parameters (learning rate, learning epochs, etc) and/or local minima, ELM avoids such issues and reaches good solutions analytically. The learning speed of ELM is extremely fast compared to other traditional methods. In this paper, we first extend the ELM algorithm from the real domain to the complex domain where the fully complex activation functions introduced by Kim and Adali [10] are used. Similar to ELM, the input weights (linking the input layer to the hidden layer) and hidden layer biases of C-ELM are randomly chosen based on some continuous distribution probablity (such as uniform distribution probability used in our simulations) and the output weights (linking the hidden layer to the output layer) are then analytically calculated. The C-ELM is used for equalization of a complex nonlinear channel with QAM signals. The simulation results show that the C-ELM equalizer is superior to CRBF [1], CMRAN [8] and CBP [9] equalizers in terms of symbol error rate (SER) and learning speed. C-ELM also avoids local minima and all the difficulties in other schemes such as tuning control parameters (learning rate, learning epochs, etc).

This paper is organized as follows. Section 2 presents the C-ELM algorithm. Section 3 shows the performance comparison of C-ELM with the CRBF, CMRAN and CBP equalizers for a QAM channel equalization problem. Discussions and conclusions are given in Section 4.

2 Complex Extreme Learning Machine (C-ELM) Algorithm

Given a series of complex-valued training samples $(\mathbf{z}_i, \mathbf{y}_i)$, $i = 1, 2, \cdots, N$, where $\mathbf{z}_i \in \mathbf{C}^n$ and $\mathbf{y}_i \in \mathbf{C}^m$, the actual outputs of the single-hidden-layer feedforward network (SLFN) with complex activation function $g_c(z)$ for these N training data is given by

$$\sum_{k=1}^{\tilde{N}} \beta_k g_c(\mathbf{w}_k \cdot \mathbf{z}_i + b_k) = \mathbf{o}_i, \quad i = 1, \cdots, N, \tag{1}$$

where column vector $\mathbf{w}_k \in \mathbf{C}^n$ is the complex input weight vector connecting the input layer neurons to the k-th hidden neuron, $\beta_k = [\beta_{k1}, \beta_{k2}, \cdots, \beta_{km}]^T \in \mathbf{C}^m$ the complex output weight vector connecting the k-th hidden neuron and the output neurons, and $b_k \in C$ is the complex bias of the k-th hidden neuron. $\mathbf{w}_k \cdot \mathbf{z}_i$ denotes the inner product of column vectors $\mathbf{w}_k$ and $\mathbf{z}_i$. g_c is a fully complex activation function.

The above N equations can be written compactly as

$$\mathbf{H}\beta = \mathbf{O} \tag{2}$$

and in practical applications the number $\tilde{N}$ of the hidden neurons is usually much less than the number N of training samples and $\mathbf{H}\beta \neq \mathbf{Y}$, where

$$\begin{aligned} &\mathbf{H}(\mathbf{w}_1, \cdots, \mathbf{w}_{\tilde{N}}, \mathbf{z}_1, \cdots, \mathbf{z}_{\tilde{N}}, b_i, \cdots, b_{\tilde{N}}) \\ &= \begin{bmatrix} g_c(\mathbf{w}_1 \cdot \mathbf{z}_1 + b_1) & \cdots & g_c(\mathbf{w}_{\tilde{N}} \cdot \mathbf{z}_1 + b_{\tilde{N}}) \\ \vdots & \cdots & \vdots \\ g_c(\mathbf{w}_1 \cdot \mathbf{z}_N + b_1) & \cdots & g_c(\mathbf{w}_{\tilde{N}} \cdot \mathbf{z}_N + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}} \end{aligned} \tag{3}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}{}^T \end{bmatrix}_{\tilde{N} \times m}, \quad \mathbf{O} = \begin{bmatrix} \mathbf{o}_1^T \\ \vdots \\ \mathbf{o}_N^T \end{bmatrix}_{N \times m} \quad \text{and} \quad \mathbf{Y} = \begin{bmatrix} \mathbf{y}_1^T \\ \vdots \\ \mathbf{y}_N^T \end{bmatrix}_{N \times m} \tag{4}$$

The complex matrix $\mathbf{H}$ is called the hidden layer output matrix. Using the analysis similar to that of ELM [6,7] and using the proof given in ([13] p.252 and [2] Theorem 2.1) we can easily show that the input weights $\mathbf{w}_i$ and hidden layer biases b_i of the SLFNs with complex activation functions (which are infinitely differentiable) can be randomly chosen and fixed based on some continuous distribution probablity instead of been trivially tuned. [1] As analyzed by Huang et. al. [6,7] for fixed input weights $\mathbf{w}_i$ and hidden layer biases b_i we can get the *least-squares solution* $\hat{\beta}$ of the linear system $\mathbf{H}\beta = \mathbf{Y}$ with *minimum norm* of output weights β, which usually tend to have good generalization performance: (Refer to Huang et. al. [4–7] for detailed analysis.)

The resulting $\hat{\beta}$ is given by:

$$\hat{\beta} = \mathbf{H}^{\dagger}\mathbf{Y} \tag{5}$$

where complex matrix $\mathbf{H}^{\dagger}$ is the *Moore-Penrose generalized inverse* (pp. 163-169 of [11]) of complex matrix $\mathbf{H}$. Thus, ELM can be extended from the real domain to a fully complex domain in a straightforward manner. The three steps in the fully complex ELM (C-ELM) algorithm can be summarized as:

Algorithm C-ELM: Given a training set $\aleph = \{(\mathbf{z}_i, \mathbf{y}_i) | \mathbf{z}_i \in \mathbf{C}^n, \mathbf{y}_i \in \mathbf{C}^m, i = 1, \cdots, N\}$, complex activation function $g_c(z)$, and hidden neuron number $\tilde{N}$,

step 1 Randomly choose the complex input weight $\mathbf{w}_k$ and the complex bias b_k, $k = 1, \cdots, \tilde{N}$.
step 2 Calculate the complex hidden layer output matrix $\mathbf{H}$.
step 3 Calculate the complex output weight β using formula (5).

Many fully complex activation functions proposed by Kim and Adali [10] can be used in our C-ELM. These include circular functions ($\tan(z) = \frac{e^{iz}-e^{-iz}}{i(e^{iz}+e^{-iz})}$, $\sin(z) = \frac{e^{iz}-e^{-iz}}{2i}$), inverse circular functions ($\arctan(z) = \int_0^z \frac{dt}{1+t^2}$, $\arcsin(z) = \int_0^z \frac{dt}{(1-t)^{1/2}}$, $\arccos(z) = \int_0^z \frac{dt}{(1-t^2)^{1/2}}$), hyperbolic functions ($\tanh(z) = \frac{e^z-e^{-z}}{e^z+e^{-z}}$, $\sinh(z) = \frac{e^z-e^{-z}}{2}$) and inverse hyperbolic functions ($\mathrm{arctanh}(z) = \int_0^z \frac{dt}{1-t^2}$, $\mathrm{arcsinh}(z) = \int_0^z \frac{dt}{(1+t^2)^{1/2}}$), where $z \in C$.

Remark: Calculation of Moore-Penrose Generalized Inverse

Definition 2.1. (pp. 163-169 of [11]) *A matrix* $\mathbf{G}$ *is the Moore-Penrose generalized inverse of (real or complex) matrix* $\mathbf{A}$*, if* $\mathbf{AGA} = \mathbf{A}$*,* $\mathbf{GAG} = \mathbf{G}$*,* $(\mathbf{AG})^* = \mathbf{AG}$*,* $(\mathbf{GA})^* = \mathbf{GA}$.

There are several methods to calculate the Moore-Penrose generalized inverse of (real or complex) matrix. These methods may include but are not limited to orthogonal projection, orthogonalization method, iterative method, and Singular Value Decomposition (SVD) [11,12]. The orthogonalization method and iterative method have their limitations since searching and iteration are used which we wish to avoid in ELM. The orthogonal project method can be used when $\mathbf{H}^*\mathbf{H}$ is nonsingular and $\mathbf{H}^{\dagger} = (\mathbf{H}^*\mathbf{H})^{-1}\mathbf{H}^*$. However, $\mathbf{H}^*\mathbf{H}$ may not always be nonsingular or may tend to be singular in some applications and

[1] The theoretical analysis such as universal approximation capability of C-ELM is currently under investigation and will appear in a future paper.

thus orthogonal projection method may not perform well in all applications. The Singular Value Decomposition (SVD) can be generally used to calculate the Moore-Penrose generalized inverse of $\mathbf{H}$ in all cases.

3 Performance Evaluation

In this section, a well-known complex nonminimum-phase channel model introduced by Cha and Kassam[1] is used to evaluate the C-ELM equalizer performance. This equalization model is of order 3 with nonlinear distortion for 4-QAM signaling. The channel output z_n (which is also the input of the equalizer) is given by

$$\begin{aligned} z_n =& o_n + 0.1o_n^2 + 0.05o_n^3 + v_n,\ v_n \sim \mathcal{N}(0, 0.01) \\ o_n =& (0.34 - i0.27)s_n + (0.87 + i0.43)s_{n-1} + (0.34 - i0.21)s_{n-2} \end{aligned} \tag{6}$$

where $\mathcal{N}(0, 0.01)$ means the white gaussian noise (of the nonminimum-phase channel) with mean 0 and variance 0.01.

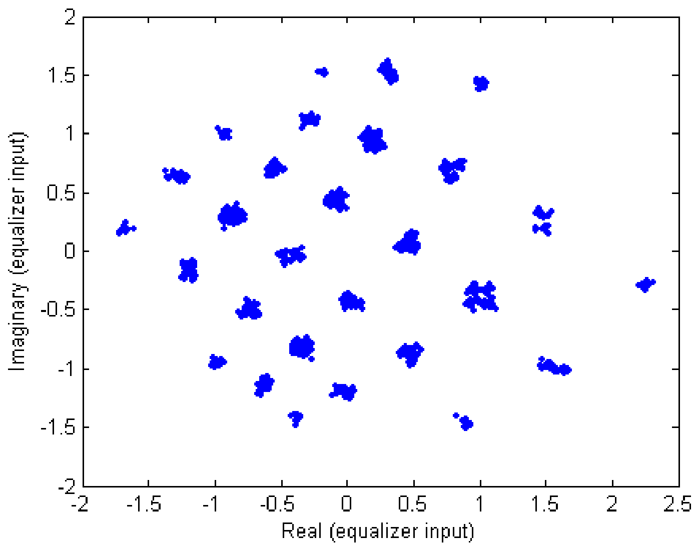

Figure 1. The distribution of the input data z_n of equalizers.

The equalizer input dimension is chosen as 3. As usually done in equalization problems, a decision delay τ is introduced in the equalizer so that at time n the equalizer estimates the input symbol $s_{n-\tau}$ rather than s_n and we set $\tau = 1$. 4-QAM symbol sequence s_n is passed through the channel and the real and imaginary parts of the symbol are valued from the set $\{\pm 0.7\}$. The fully complex activation function of both C-ELM and CBP is chosen as $\text{arcsinh}(z) = \int_0^z \frac{dt}{(1+t^2)^{1/2}}$, where $z = \mathbf{w} \cdot \mathbf{z} + b$. In fact, during our studies we find that CBP with the hyperbolic activation function $\tanh(z)$ does not converge well

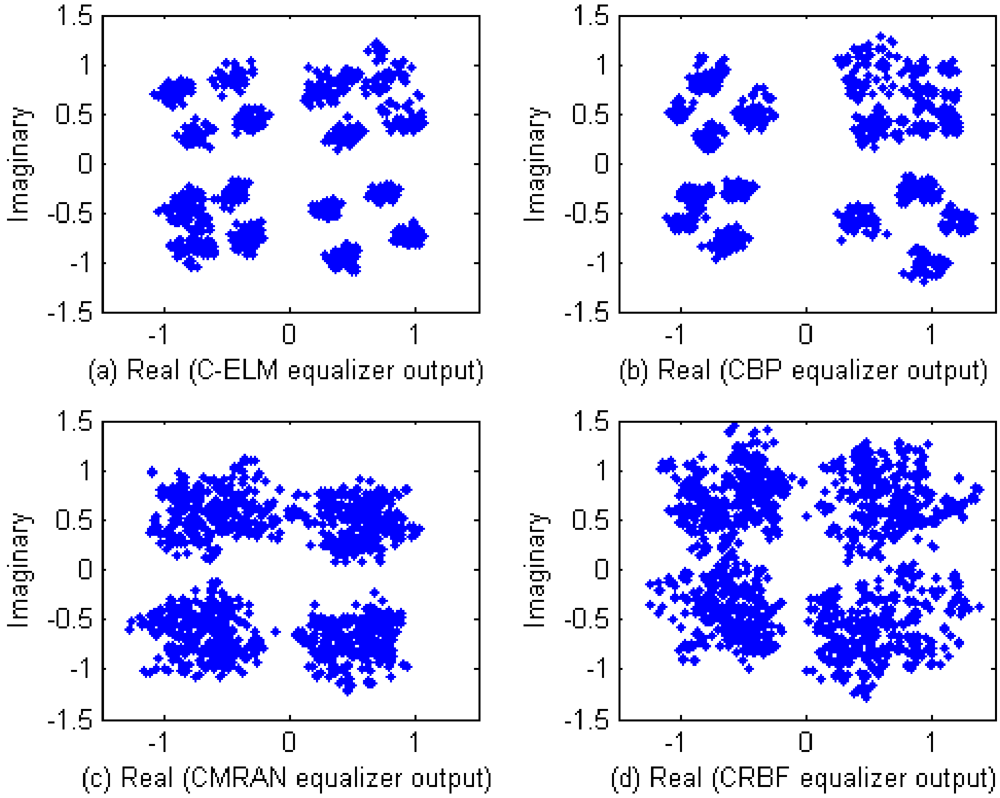

Figure 2. Eye diagram of the outputs of different equalizers (a) C-ELM, (b) CBP, (c) CMRAN, (d) CRBF.

and produce oscillation in the error but CBP with the activation function arcsinh(z) converges, however C-ELM works well with both these complex activation functions and many others. The reason may be that CBP gets stuck in local minima easily while ELM tends to reach global minimum directly. Both the input weight vectors $\mathbf{w}_k$ and biases b_k of the C-ELM [2] are randomly chosen from a complex area centered at the origin with the radius set as 0.1.

All the three equalizers: CMRAN, CBP and C-ELM are trained with 1000 data symbols at 16dB SNR. It is found that the CRBF equalizer trained with such small number of training data cannot classify the testing symbols clearly and thus a higher number (10^4) of training data are used to train CRBF equalizer. The hidden neuron numbers of C-ELM and CBP are set to 10. The CMRAN equalizer obtains 22 hidden neurons at the end of the training process after self growing and pruning neurons during training. Different numbers of hidden neurons have been tried for the CRBF equalizer, however, the optimal hidden neuron number of CRBF equalizer is found to be 30.

[2] Open source codes of the ELM algorithm with different testing cases can be downloaded from: http://www.ntu.edu.sg/eee/icis/cv/egbhuang.htm

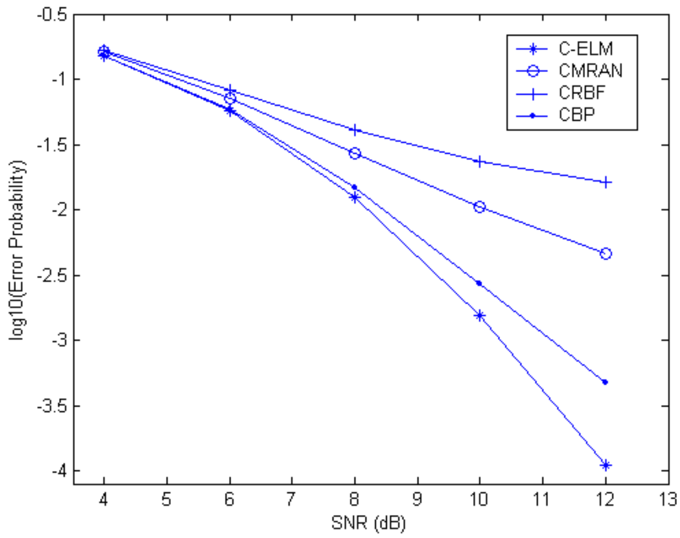

Figure 3. Error probability for C-ELM, CMRAN, CRBF and CBP

All the simulations are conducted in a MATLAB environment running in an ordinary PC with 3GHZ CPU. Figure 1 shows the distribution of the input data of the different equalizers and Figure 2 shows the eye diagram of the outputs of the four neural equalizers: C-ELM, CBP, CMRAN and CRBF, respectively. As observed from Figure 2 both C-ELM and CBP can separate the outputs into four regions clearly. Average of 10^6 testing samples at various SNRs were used for computing the symbol error rate (SER) and the comparison of SER for all the four equalizers is shown in Figure 3. As observed from Figure 3, C-ELM is superior to all other equalizers in terms of SER. Table 1 shows the training and testing time comparison for the four equalizers. It can be seen that the C-ELM equalizer can complete training much faster than all other equalizers.

Algorithms	Neurons	Number of training data	Training time (s)	Speedup
C-ELM	10	1000	0.032	1
CBP	10	1000	1.266	39.56
CMRAN	22	1000	25.481	796.28
CRBF	30	10^4	46.331	1447.84

Table 1
Time comparisons of the four equalizers (a) C-ELM, (b) CBP, (c) CMRAN, (d) CRBF.

4 Discussions and Conclusions

In this paper, we propose a fully complex learning algorithm for single-hidden-layer feedforward neural networks (SLFNs) which is referred to as fully complex extreme learning machine (C-ELM) and its performance has been tested in communication channel equalizers. Similar to ELM[7,6], the input weights (linking the input layer to the hidden layer) and hidden layer biases of C-ELM are randomly generated and then the output weights (linking the hidden layer to the output layer) are simply analytically calculated instead of iteratively tuned. As observed from the simulation results, the proposed C-ELM can complete the learning phase in an extremely fast speed and obtain much lower symbol error rate (SER). Consistent to the conclusion of Kim and Adali [10] compared to split-complex activation (basis) functions based on neural models (CMRAN and CRBF) the fully complex models (C-ELM and CBP) provide parsimonious structures for applications in the complex domain. It should be noted that as analyzed by Kim and Adali [10] the CBP learning algorithm is sensitive to the size of the learning rate and the radius of initial random weights [3] and as done in our simulations the learning rate and the radius of initial random weights need to be carefully tuned. Different from other equalizers, C-ELM has avoided well the difficulties in manually tuning control parameters (learning rate, initial weights/biases, learning epochs) and prevented local minima by reaching the good solutions analytically. C-ELM can be implemented and used easily. In fact, faster learning speed, faster response and ease of implementation are key to the success of the communication channel equalizers. In principle, as tested in our various simulations, many fully complex activation functions introduced by Kim and Adali [10] can be used in the proposed C-ELM and its universal approximation capability will be provided in details in the near future.

References

[1] I. Cha and S. A. Kassam. Channel equalization using adaptive complex radial basis function networks. *IEEE Journal Selected Areas in Communications*, 13(1):122–131, January 1995.

[2] G.-B. Huang. Learning capability and storage capacity of two-hidden-layer feedforward networks. *IEEE Transactions on Neural Networks*, 14(2):274 –281, March 2003.

[3] G.-B. Huang, Y.-Q. Chen, and H. A. Babri. Classification ability of single

[3] We would like to thank T. Kim for further clarification on the sensitivity and convergence of CBP in our private communications.

hidden layer feedforward neural networks. *IEEE Transactions on Neural Networks*, 11(3):799–801, May 2000.

[4] G.-B. Huang and C.-K. Siew. Extreme learning machine with randomly assigned RBF kernels. *International Journal of Information Technology*, 11(1), March 2005.

[5] G.-B. Huang and C.-K. Siew. Extreme learning machine: RBF network case. In *Proceedings of the Eighth International Conference on Control, Automation, Robotics and Vision (ICARCV 2004)*, Kunming, China, 6-9 Dec, 2004.

[6] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew. Extreme learning machine: Theory and applications. *Submitted to Neurocomputing*, 2005.

[7] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew. Extreme learning machine: A new learning scheme of feedforward neural networks. In *Proceedings of International Joint Conference on Neural Networks (IJCNN2004)*, Budapest, Hungary, 25-29 July, 2004.

[8] D. Jianping, N. Sundararajan, and P. Saratchandran. Communication channel equalization using complex-valued minimal radial basis function neural networks. *IEEE Transactions on Neural Networks*, 13(3):687–696, May 2002.

[9] T. Kim and T. Adali. Complex backpropagation neural network using elementary transcendental activation functions. In *Proceedings of IEEE International Conference on Acoustics, Speech, Signal Processing (ICASSP)*, volume 2, pages 1281–1284, Salt Lake City, UT, May 2001.

[10] T. Kim and T. Adali. Approximation by fully complex multilayer perceptrons. *Neural Computation*, 15(7):1641–1666, July 2003.

[11] J. M. Ortega. *Matrix Theory.* Plenum Press, New York and London, 1987.

[12] C. R. Rao and S. K. Mitra. *Generalized Inverse of Matrices and its Applications.* New York: John Wiley, 1971.

[13] S. Tamura and M. Tateishi. Capabilities of a four-layered feedforward neural network: Four layers versus three. *IEEE Transactions on Neural Networks*, 8(2):251–255, March 1997.