Is a Complex-Valued Stepsize Advantageous in Complex-Valued Gradient Learning Algorithms?

Huisheng Zhang and Danilo P. Mandic, *Fellow, IEEE*

***Abstract*— Complex gradient methods have been widely used in learning theory, and typically aim to optimize real-valued functions of complex variables. The stepsize of complex gradient learning methods (CGLMs) is a positive number, and little is known about how a complex stepsize would affect the learning process. To this end, we undertake a comprehensive analysis of CGLMs with a complex stepsize, including the search space, convergence properties, and the dynamics near critical points. Furthermore, several adaptive stepsizes are derived by extending the Barzilai–Borwein method to the complex domain, in order to show that the complex stepsize is superior to the corresponding real one in approximating the information in the Hessian. A numerical example is presented to support the analysis.**

***Index Terms*— Barzilai–Borwein method (BBM), complex gradient method, complex stepsize, complex-valued neural networks (CVNNs), convergence.**

I. Introduction

Complex-valued neural networks (CVNNs) have attracted widespread interest in a variety of disciplines, including array signal processing, radar and magnetic resonance data imaging, communication systems, and interval data processing [1]–[4]. The popularity of CVNNs stems not only from the mathematical advantages over the bivariate reals when dealing with complex-valued signals but also owing to several special properties of CVNNs, which make them more powerful than the traditional real-valued neural networks (RVNNs). For example, the orthogonality of the decision boundary greatly enhances the generalization ability of CVNNs [5], while the widely linear processing can capture the complete second-order statistical relationship between the input and the output [1], [6], [7].

Based on the different choices of the activation functions, the three typical feedforward CVNN models are: 1) the real-imaginary CVNN [8]; 2) the amplitude-phase CVNN [9]; and 3) the fully CVNN [10], [11]. Gradient training algorithms for these networks have been originally proposed in [8]–[12]; however, most of these algorithmic expressions appear cluttered, as the real part and the imaginary part (or the amplitude and the phase) of the network parameters are treated separately. To overcome this drawback, Wirtinger calculus (also known as the CR calculus) [13]–[15] has been introduced for the optimization of real-valued cost functions with respect to complex variables [1], [16]. With the help of Wirtinger calculus, several elegant and efficient gradient training methods for fully CVNNs have been proposed [17], [18]. In addition, the gradient training methods for real-imaginary CVNNs and amplitude-phase CVNNs can also be derived under the framework of Wirtinger calculus [19]. In this brief, we refer to this class of gradient learning algorithms for CVNNs as complex gradient learning methods (CGLMs).

Similar to its real counterpart, CGLMs suffer from slow convergence due to the high nonconvexity of the cost function and the numerous plateaus around its saddle points. For a given CVNN architecture, the convergence rate of CGLMs depends critically on the stepsize and the search direction, which are usually set as a small positive number and the negative gradient direction, respectively. It has been shown that the adaptive stepsize [20], [21] and the search direction based on the second derivative information [22] can greatly accelerate the convergence. Moreover, Kim and Adali [10], [11] proposed to use an imaginary or a complex stepsize; however, they found no significant difference in performance. More recently, the enhanced performance of a complex stepsize has been experimentally shown for complex-valued tensor decomposition problems [23]. However, both the theoretical and experimental justifications for the advantages and disadvantages of the complex stepsize for CGLMs are still lacking.

This brief aims to provide insights into the essential nature of the CGLMs with a complex stepsize. Our contributions are as follows.

1) We show that the complex stepsize extends the search space of CGLMs from a half-line to a half-plane, which equips the learning process with more degrees of freedom; moreover, it is shown that the complex stepsize algorithms offer enhanced ability to escape from saddle points.
2) Convergence results for CGLMs with a complex stepsize are established; the analysis shows that a pure imaginary number should not be used as the stepsize of CGLMs.
3) By analyzing the dynamics of CGLMs with a complex stepsize near the critical points, several circumstances where a constant complex stepsize should be avoided are identified.
4) An adaptive complex stepsize strategy for CGLMs is proposed by extending the Barzilai–Borwein method (BBM) [24], [25] to the complex domain, and it is shown that the generic extensions of BBM to the complex domain should employ a complex stepsize.

The remainder of this brief is organized as follows. In Section II, we list notations and provide a background on Wirtinger calculus. Section III presents some heuristics and theoretical analysis for CGLM with a complex stepsize, followed by several adaptive complex stepsize strategies for CGLMs in Section IV. A numerical example is given in Section V to support the analysis. A summary of the findings and the conclusions are given in Section VI.

II. Notations and Preliminary

A. Notations

In this brief, the following notations are adopted.

1) Bold-faced quantities with uppercase and lowercase letters denote, respectively, matrices and vectors.
2) $\mathbb{R}$ denotes the set of real numbers.

Manuscript received March 27, 2015; revised October 18, 2015; accepted October 21, 2015. Date of publication November 5, 2015; date of current version November 15, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant 61101228 and Grant 61402071, in part by the Liaoning Provincial Natural Science Foundation of China under Grant 2015020011, and in part by the Fundamental Research Funds for the Central Universities of China under Grant 3132015157.

H. Zhang is with the Department of Mathematics, Dalian Maritime University, Dalian 116026, China, and also with the Electrical and Electronic Engineering Department, Imperial College London, London SW7 2BT, U.K. (e-mail: zhhuisheng@163.com).

D. P. Mandic is with the Electrical and Electronic Engineering Department, Imperial College London, London SW7 2BT, U.K. (e-mail: d.mandic@imperial.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2015.2494361

3) $\mathbb{C}$ denotes the set of complex numbers.
4) $|\cdot|$ denotes the absolute value of a variable.
5) $\|\cdot\|$ denotes the Euclidean norm of a vector.
6) $(\cdot)^*$ denotes the complex conjugate.
7) $(\cdot)^T$ denotes the transpose of a vector or a matrix.
8) $(\cdot)^H$ denotes the Hermitian transpose of a vector or a matrix.
9) $\Re(\cdot)$ denotes the real part of a complex number or a vector.
10) $\Im(\cdot)$ denotes the imaginary part of a complex number or a vector.

B. Wirtinger Calculus

Consider a function $f(z): \mathbb{C} \to \mathbb{C}$, given by

$$f(z) = u(x, y) + iv(x, y) \tag{1}$$

where $i = \sqrt{-1}$ and $z = x + iy$. Upon substituting

$$x = \frac{z + z^*}{2}, \quad y = \frac{z - z^*}{2i} \tag{2}$$

the mapping f can be rewritten as a bivariate function of z and z^*. If f is real differentiable, i.e., $u(x, y)$ and $v(x, y)$ are differentiable with respect to real-valued variables x and y, we can then apply Wirtinger calculus to compute $(\partial f/\partial z)$ and $(\partial f/\partial z^*)$ by treating z and z^* as two independent variables. In this way, when taking the partial derivative with respect to z, we consider z^* as a constant and calculate $(\partial f/\partial z)$. Similarly, $(\partial f/\partial z^*)$ is derived by considering z as a constant and taking the partial derivative with respect to z^*. This makes it possible for $(\partial f/\partial z)$ and $(\partial f/\partial z^*)$ to be derived in the same manner as for the real-valued case. Upon applying the chain rule, we obtain

$$\begin{aligned} \frac{\partial f}{\partial z} &= \frac{1}{2}\left(\frac{\partial f}{\partial x} - i\frac{\partial f}{\partial y}\right) \\ \frac{\partial f}{\partial z^*} &= \frac{1}{2}\left(\frac{\partial f}{\partial x} + i\frac{\partial f}{\partial y}\right). \end{aligned} \tag{3}$$

If function f is real-valued, i.e., $v(x, y) = 0$, based on (3), it is straightforward to obtain the following property:

$$\left(\frac{\partial f}{\partial z}\right)^* = \frac{\partial f}{\partial z^*}. \tag{4}$$

If $v(x, y) \neq 0$, a more general result is obtained

$$\left(\frac{\partial f}{\partial z}\right)^* = \frac{\partial f^*}{\partial z^*}. \tag{5}$$

Wirtinger calculus not only offers an elegant way to compute the two partial derivatives $(\partial f/\partial z)$ and $(\partial f/\partial z^*)$, but also provides a solution to the optimization of real-valued cost functions of complex-valued variables. This is particularly important for learning machines, as the negative of the gradient with respect to the conjugate of the network parameters defines the direction of the steepest descent; this is key in designing gradient-based optimization algorithms for the minimization of real-valued cost functions with respect to complex-valued variables.

III. Complex Gradient Learning Method With Complex Stepsize

A. Algorithm Description

For the convenience of presentation, we consider a single output complex-valued feedforward neural network model with weight vector $\mathbf{w} \in \mathbb{C}^N$.

Suppose that $\{(\boldsymbol{\xi}^k, d^k)\}_{k=1}^K \subset \mathbb{C}^p \times \mathbb{C}$ is a given set of training samples, where $\boldsymbol{\xi}^k$ is the input, and d^k is the corresponding desired output. The aim of the network training is to find the optimal weight vector $\mathbf{w}^\bigstar$, which minimizes the error function

$$E = \sum_{k=1}^{K} (d^k - o(\boldsymbol{\xi}^k, \mathbf{w}))(d^k - o(\boldsymbol{\xi}^k, \mathbf{w}))^* = \sum_{k=1}^{K} e_k e_k^* \tag{6}$$

where $o(\boldsymbol{\xi}^k, \mathbf{w})$ denotes the output of the network for the input $\boldsymbol{\xi}^k$ and $e_k = d^k - o(\boldsymbol{\xi}^k, \mathbf{w})$ is the output error for the training sample $(\boldsymbol{\xi}^k, d^k)$.

From (6), the error function E can be viewed as a function of complex variable $\mathbf{w}$ and its complex conjugate $\mathbf{w}^*$, and for brevity, we shall use $E(\mathbf{w})$ to denote this functional relationship. As the gradient $\nabla_{\mathbf{w}^*} E$ defines the direction of the maximum rate of change with respect to $\mathbf{w}$, starting from an initial point $\mathbf{w}^{(0)}$ [14], the cost function $E(\mathbf{w})$ can be minimized using complex gradient descent in the following recursive form:

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \eta_k \nabla_{\mathbf{w}^*} E(\mathbf{w}^{(k)}) \tag{7}$$

where η_k is the stepsize at the kth iteration, which is usually set to be a small positive number.

For a complex-valued η_k, (7) takes the following form:

$$\begin{aligned} \mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} &- (\Re(\eta_k)\nabla_{\mathbf{w}^*} E(\mathbf{w}^{(k)}) \\ &+ i\Im(\eta_k)\nabla_{\mathbf{w}^*} E(\mathbf{w}^{(k)})). \end{aligned} \tag{8}$$

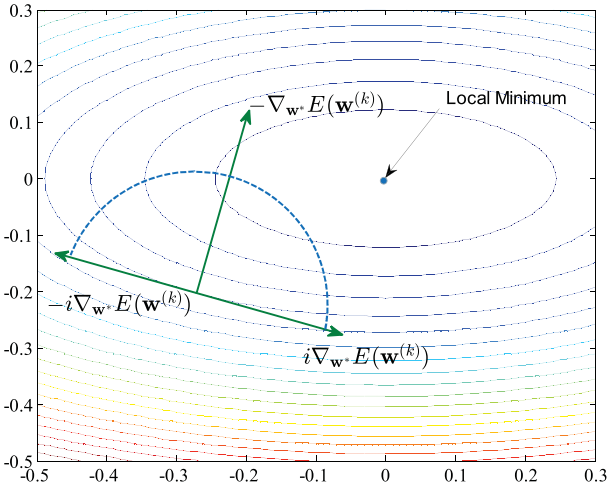

Fig. 1. Illustration of the search space for CGLMs with complex stepsize.

Fig. 1 shows that, geometrically, the vector $i\nabla_{\mathbf{w}^*} E(\mathbf{w}^{(k)})$ is rotated with respect to $-\nabla_{\mathbf{w}^*} E(\mathbf{w}^{(k)})$ clockwise, by an angle $(\pi/2)$. Thus, the increment $-\eta_k \nabla_{\mathbf{w}^*} E(\mathbf{w}^{(k)})$ lives in a half-plane spanned by the two orthogonal vectors $-\nabla_{\mathbf{w}^*} E(\mathbf{w}^{(k)})$ and $i\nabla_{\mathbf{w}^*} E(\mathbf{w}^{(k)})$, for all $\eta_k \in \mathbb{C}$ with $\Re(\eta_k) > 0$. Therefore, the complex stepsize extends the search space from a half-line (for a standard positive stepsize) to a half-plane, hence providing more freedom in optimization.

Remark 1: If the imaginary part of the stepsize is nonzero, CGLM with a complex stepsize does not follow the usual steepest descent direction, i.e., the negative of the gradient. However, there is plenty of evidence that the negative of the gradient may not be the best or the actual steepest descent direction in iterative learning. For example, some variants of standard gradient descent method, such as the conjugate gradient method and the gradient method with momentum, do not update the parameters in the steepest descent direction, but they converge faster than the standard gradient descent. Therefore, it comes as no surprise that the gradient descent with a complex stepsize gradient descent exhibits similar behavior.

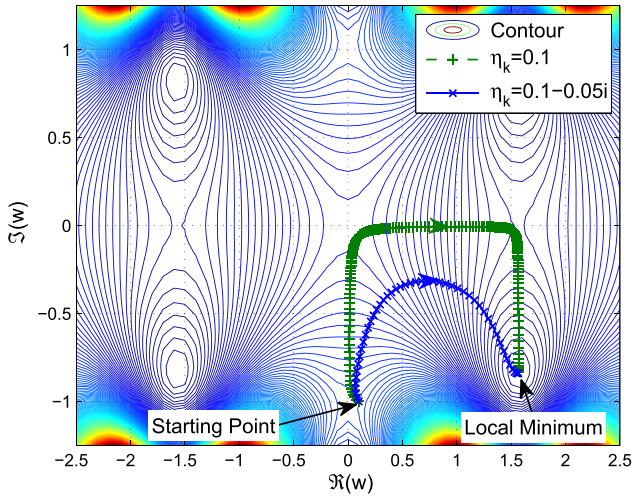

Fig. 2. Learning trajectories for CGLM with the complex stepsize against the real stepsize on a contour map.

B. Ability to Escape the Saddle Points

Gradient-based learning algorithms converge to points with vanishing gradient; such points are referred to as critical points and include both the saddle points and local minima. Nitta [26] pointed out a prominent property of CVNNs—owing to the hierarchical structure of complex-valued neural networks most of their critical points are saddle points—critical points of RVNNs tend to be local minima. It is important to notice that as saddle point is unstable, in most cases, the algorithm will eventually converge to a local minimum. This property points to an advantage of CVNNs. During the gradient training process, they are more likely to reach a global minimum than the RVNNs. However, this property also poses a challenge for complex gradient training algorithms, as too many saddle points will greatly slow down the convergence [27]. Thus, it is of great importance for a learning algorithm to be able to automatically escape the saddle points.

In the sequel, we show that the complex stepsize is well equipped to address this issue. When a real gradient learning algorithm with a small positive stepsize is stuck in the plateau of a saddle point, the algorithm moves very slowly along the direction of the negative gradient. By including an imaginary part in the stepsize, a new search direction is obtained, which is a linear combination of the negative gradient vector $-\nabla_{\mathbf{w}^*}E(\mathbf{w}^{(k)})$ and its orthogonal vector $i\nabla_{\mathbf{w}^*}E(\mathbf{w}^{(k)})$, thus offering another degree of freedom to escape the saddle point. To illustrate this concept, consider the function

$$(\cos(\cos(\cos(w))) - 0.1)(\cos(\cos(\cos(w))) - 0.1)^*, \quad w \in \mathbb{C} \tag{9}$$

which mimics the error function of a hierarchical CVNN. Starting from an initial point $0.1 - i$, Fig. 2 shows the contour plot of the function in (9) and the trajectories for the learning process of the complex gradient algorithm with: 1) a real stepsize 0.1 and 2) a complex stepsize $0.1 - 0.05i$. Notice that the learning process with a real stepsize slowed down at the saddle point, whereas the learning process with a complex stepsize successfully avoided the saddle point, and converged to a minimum faster than the CGLM with a real stepsize.

C. Convergence Analysis

It has been proved that the convergence of gradient descent based on $\nabla_{\mathbf{w}^*}E(\mathbf{w})$ can be guaranteed if η_k is a sufficiently small positive number [28]. In the following theorem, we show that the complex stepsize can also guarantee the decrease in $E(\mathbf{w})$ along iterations and the convergence of $\nabla_{\mathbf{w}^*}E(\mathbf{w})$.

Theorem 2: Let the sequence $\{\mathbf{w}^{(k)}\}$ be generated by (7) starting from an arbitrary initial value $\mathbf{w}^{(0)}$, and let $\nabla_{\mathbf{w}^*}E(\mathbf{w})$ satisfy the Lipschitz condition, that is, there exists a positive constant L, such that

$$||\nabla_{\mathbf{w}^*}E(\mathbf{w}_1) - \nabla_{\mathbf{w}^*}E(\mathbf{w}_2)|| \le L||\mathbf{w}_1 - \mathbf{w}_2|| \tag{10}$$

for any $\mathbf{w}_1 \in \mathbb{C}^N$ and $\mathbf{w}_2 \in \mathbb{C}^N$. Then, if the stepsize η_k satisfies

$$\begin{aligned}\frac{1-\sqrt{1-4L^2\theta^2(\Im(\eta_k))^2}}{2L\theta} &< \Re(\eta_k) \\ &< \frac{1+\sqrt{1-4L^2\theta^2(\Im(\eta_k))^2}}{2L\theta}\end{aligned} \tag{11}$$

where θ is a constant in the interval $(0, 1)$, the following holds: $E(\mathbf{w}^{(k+1)}) < E(\mathbf{w}^{(k)})$ and $\lim_{k\to\infty}\nabla_{\mathbf{w}^*}E(\mathbf{w}^{(k)}) = \mathbf{0}$.

Proof: As $E(\mathbf{w})$ is real-valued, using (4), we have

$$\nabla_{\mathbf{w}^*}E(\mathbf{w}) = (\nabla_{\mathbf{w}}E(\mathbf{w}))^*. \tag{12}$$

Now, using (7), (10), and (12), the mean value theorem and the triangle inequality, we have

$$\begin{aligned}
&E(\mathbf{w}^{(k+1)}) - E(\mathbf{w}^{(k)}) \\
&\quad= (\nabla_{\mathbf{w}}E(\mathbf{w}^{(k)} + \theta(\mathbf{w}^{(k+1)} - \mathbf{w}^{(k)})))^T(\mathbf{w}^{(k+1)} - \mathbf{w}^{(k)}) \\
&\qquad+ (\nabla_{\mathbf{w}^*}E(\mathbf{w}^{(k)} + \theta(\mathbf{w}^{(k+1)} - \mathbf{w}^{(k)})))^T(\mathbf{w}^{(k+1)} - \mathbf{w}^{(k)})^* \\
&\quad= (\nabla_{\mathbf{w}}E(\mathbf{w}^{(k)}))^T(\mathbf{w}^{(k+1)} - \mathbf{w}^{(k)}) \\
&\qquad+ (\nabla_{\mathbf{w}^*}E(\mathbf{w}^{(k)}))^T(\mathbf{w}^{(k+1)} - \mathbf{w}^{(k)})^* \\
&\qquad+ (\nabla_{\mathbf{w}}E(\mathbf{w}^{(k)} + \theta(\mathbf{w}^{(k+1)} - \mathbf{w}^{(k)})) - \nabla_{\mathbf{w}}E(\mathbf{w}^{(k)}))^T \\
&\qquad\times (\mathbf{w}^{(k+1)} - \mathbf{w}^{(k)}) \\
&\qquad+ (\nabla_{\mathbf{w}^*}E(\mathbf{w}^{(k)} + \theta(\mathbf{w}^{(k+1)} - \mathbf{w}^{(k)})) - \nabla_{\mathbf{w}^*}E(\mathbf{w}^{(k)}))^T \\
&\qquad\times (\mathbf{w}^{(k+1)} - \mathbf{w}^{(k)})^* \\
&\quad\le 2\Re((\nabla_{\mathbf{w}}E(\mathbf{w}^{(k)}))^T(\mathbf{w}^{(k+1)} - \mathbf{w}^{(k)})) \\
&\qquad+ 2\|\nabla_{\mathbf{w}}E(\mathbf{w}^{(k)} + \theta(\mathbf{w}^{(k+1)} - \mathbf{w}^{(k)})) - \nabla_{\mathbf{w}}E(\mathbf{w}^{(k)})\| \\
&\qquad\times \|\mathbf{w}^{(k+1)} - \mathbf{w}^{(k)}\| \\
&\quad\le -2\Re(\eta_k(\nabla_{\mathbf{w}^*}E(\mathbf{w}^{(k)}))^H(\nabla_{\mathbf{w}^*}E(\mathbf{w}^{(k)}))) \\
&\qquad+ 2L\theta\|\mathbf{w}^{(k+1)} - \mathbf{w}^{(k)}\|^2 \\
&\quad= 2(-\Re(\eta_k) + L\theta|\eta_k|^2)\|\nabla_{\mathbf{w}^*}E(\mathbf{w}^{(k)})\|^2
\end{aligned} \tag{13}$$

where θ is a constant in $(0, 1)$. In order to verify $E(\mathbf{w}^{(k+1)}) < E(\mathbf{w}^{(k)})$, we only require the stepsize η_k to satisfy

$$-\Re(\eta_k) + L\theta|\eta_k|^2 < 0 \tag{14}$$

which is equivalent to (11). Let $\gamma = 2(\Re(\eta_k) - L\theta|\eta_k|^2)$, then using (13), we have

$$\begin{aligned}E(\mathbf{w}^{(k+1)}) &\le E(\mathbf{w}^{(k)}) - \gamma\|\nabla_{\mathbf{w}^*}E(\mathbf{w}^{(k)})\|^2 \\ &\le \cdots \le E(\mathbf{w}^{(0)}) - \gamma\sum_{t=0}^{k}\|\nabla_{\mathbf{w}^*}E(\mathbf{w}^{(t)})\|^2.\end{aligned} \tag{15}$$

Since $E(\mathbf{w}) \ge 0$, from (15), it then follows that $\sum_{t=0}^{+\infty}\|\nabla_{\mathbf{w}^*}E(\mathbf{w}^{(t)})\|^2 < +\infty$, which implies $\lim_{k\to\infty}\nabla_{\mathbf{w}^*}E(\mathbf{w}^{(k)}) = \mathbf{0}$. This completes the proof. □

Remark 3: Theorem 2 establishes the bounds on the complex stepsize in order to ensure the decrease in the error function along the iterations and the convergence of its gradient. We should mention that, as shown in [29, Lemma 4.2] and [30, Lemma 2], the Lipschitz condition in Theorem 2 and the value L do exist for a variety of activation functions, including both locally analytic functions and traditional split-complex functions. From (11), it is important to point out that, although the real part of the stepsize must be positive,

there is no sign restriction for the imaginary part. This also means that the positive real part of the stepsize is a necessary condition for the algorithm to converge; in other words, pure imaginary numbers should not be used as the stepsize.

D. Dynamics Near the Critical Points

We next investigate the dynamics of complex gradient methods with a constant complex stepsize near a critical point, in order to identify several application scenarios where the constant complex stepsize should be avoided.

The complex gradient method (7) can be viewed as a discrete dynamical system, given by

$$\mathbf{w}^{(k+1)} = \mathbf{g}(\mathbf{w}^{(k)}) \tag{16}$$

where the vector-valued function $\mathbf{g}(\mathbf{w}) = \mathbf{w} - \eta\nabla_{\mathbf{w}^*}E(\mathbf{w})$. Notice that the fixed point of the dynamical system in (16) is precisely the critical point of $E(\mathbf{w})$.

Suppose $\mathbf{w}^\bigstar$ is a fixed point of (16), and

$$\mathbf{g}'(\mathbf{w}) = \mathbf{I} - \eta\nabla_{\mathbf{w}^*\mathbf{w}^T}E(\mathbf{w}) \tag{17}$$

is the Jacobian matrix of $\mathbf{g}(\mathbf{w})$, where $\mathbf{I}$ denotes the identity matrix of an appropriate order. Then, $\mathbf{w}^\bigstar$ is an attractor of (16) if the spectral radius (maximum modulus of the eigenvalues of a matrix) $\rho(\mathbf{g}'(\mathbf{w}^\bigstar)) < 1$. The smaller $\rho(\mathbf{g}'(\mathbf{w}^\bigstar))$, the faster the convergence of the sequence $\{\mathbf{w}^{(k)}\}$ generated by (16) from an initial point close enough to $\mathbf{w}^\bigstar$.

As $\nabla_{\mathbf{w}^*\mathbf{w}^T}E(\mathbf{w}^\bigstar)$ is a Hermitian symmetric matrix, all its eigenvalues are real. Suppose λ is an eigenvalue of $\nabla_{\mathbf{w}^*\mathbf{w}^T}E(\mathbf{w}^\bigstar)$, then

$$\begin{aligned}|1-\eta\lambda|^2 &= |1-(\Re(\eta)+i\Im(\eta))\lambda|^2\\ &= (1-\Re(\eta)\lambda)^2+(\Im(\eta)\lambda)^2\\ &\geq (1-\Re(\eta)\lambda)^2\end{aligned} \tag{18}$$

to yield

$$\rho(\mathbf{g}'(\mathbf{w}^\bigstar)) \geq \rho(\mathbf{I}-\Re(\eta)\nabla_{\mathbf{w}^*\mathbf{w}^T}E(\mathbf{w}^\bigstar)). \tag{19}$$

From the above, we can make the following observations regarding the choice of the stepsize, given in Remarks 4–7.

Remark 4: From (19), we can conclude that when $\mathbf{w}^{(k)}$ is close enough to $\mathbf{w}^\bigstar$, for which $E(\mathbf{w}^\bigstar)$ is a local minimum, and a constant stepsize is used during the learning process, the algorithm with a constant complex stepsize η will converge slower and may be less stable than when using a real stepsize $\Re(\eta)$. The slower convergence can also be explained from the geometry viewpoint: for a complex constant stepsize, $\mathbf{w}^{(k)}$ will rotate around the minimum point, thus slowing down the convergence. On the other hand, if the critical point is not a minimum but a saddle point, the learning algorithm with a complex stepsize offers enhanced ability to escape from the attraction of this saddle point. This conforms with our analysis in Section III-B.

Remark 5: Suppose $E(\mathbf{w})$ is a quadratic function of the form

$$E(\mathbf{w}) = c + \mathbf{b}^H\mathbf{w} + \mathbf{b}^T\mathbf{w}^* + \mathbf{w}^H\mathbf{A}\mathbf{w} \tag{20}$$

where $c \in \mathbb{R}$, $\mathbf{b} \in \mathbb{C}^n$, and $\mathbf{A}$ is a positive definite (or semidefinite) Hermitian matrix. Then, the complex gradient method in (7) takes the form

$$\begin{aligned}\mathbf{w}^{(k+1)} &= \mathbf{w}^{(k)} - \eta(\mathbf{A}\mathbf{w}^{(k)} + \mathbf{b})\\ &= (\mathbf{I}-\eta\mathbf{A})\mathbf{w}^{(k)} - \eta\mathbf{b}.\end{aligned} \tag{21}$$

Similarly to (19), we now have $\rho(\mathbf{I}-\eta\mathbf{A}) \geq \rho(\mathbf{I}-\Re(\eta)\mathbf{A})$. Thus, in this case, the constant real stepsize is superior to the constant complex stepsize.

Remark 6: Consider a complex linear feedforward filter with the mean square error to be minimized, given by

$$\begin{aligned}\mathbb{E}[e(k)e^*(k)] &= \mathbb{E}[(d(k)-\mathbf{w}^H\mathbf{x}(k))(d(k)-\mathbf{w}^H\mathbf{x}(k))^*]\\ &= \mathbb{E}[d(k)d^*(k) - \mathbf{w}^H\mathbf{x}(k)d^*(k)\\ &\quad - \mathbf{w}^T\mathbf{x}^*(k)d(k) + \mathbf{w}^H\mathbf{x}(k)\mathbf{w}^T\mathbf{x}^*(k)]\\ &= \mathbb{E}[|d(k)|^2] - \mathbf{w}^H\mathbb{E}[\mathbf{x}(k)d^*(k)]\\ &\quad - \mathbf{w}^T\mathbb{E}[\mathbf{x}^*(k)d(k)] + \mathbf{w}^H\mathbb{E}[\mathbf{x}(k)\mathbf{x}^H(k)]\mathbf{w}\end{aligned} \tag{22}$$

where the symbol $\mathbb{E}[\cdot]$ denotes the statistical expectation operator, $d(k)$ and $\mathbf{x}(k)$ are, respectively, the desired output and input vector at iteration k, and $\mathbf{w}$ is the weight vector. This error function is obviously a quadratic function of the form (20). If the stepsize is a very small constant, the dynamics of this filter are very similar to the system in (16), and there is no advantage of a complex-valued constant stepsize.

Remark 7: The above discussion shows that, when the iteration $\mathbf{w}^{(k)}$ is very close to an optimal $\mathbf{w}^\bigstar$, or the objective function is quadratic of the form (20), the constant complex stepsize is not a better choice than the constant real stepsize. However, this does not prevent us from using the complex stepsize even in the above situations. With an appropriate adaptive strategy, the complex stepsize could still outperform the real one, as shown in Section IV.

IV. Complex Barzilai–Borwein Training Method

The BBM [24] is an efficient strategy for adaptively choosing the stepsize of gradient algorithms. With very low computation and memory requirements, the BBM is an alternative to conjugate gradient methods [25], and has been used in a variety of applications [31]. In this section, we extend the BBM to train CVNNs, by deriving the complex Barzilai–Borwein learning method with both a complex stepsize (CBBM-CSS) and a real stepsize (CBBM-RSS). We also show that the CBBM-CSS has a generic form and is superior to CBBM-RSS.

The well-known Newton method is defined by

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - (\mathbf{H}^{(k)})^{-1}\nabla_{\mathbf{w}^*}E(\mathbf{w}^{(k)}) \tag{23}$$

where $\mathbf{H}^{(k)}$ is the Hessian matrix of $E(\mathbf{w})$ at the point $\mathbf{w} = \mathbf{w}^{(k)}$. As the computation of the Hessian matrix and its inverse is very time-consuming, it is advantageous to use a simpler matrix to approximate $\mathbf{H}^{(k)}$.

Motivated by the following relationship between the second derivative and the first derivative:

$$f''(z_0)(z-z_0) \approx f'(z) - f'(z_0) \tag{24}$$

when z is very close to z_0, then the quasi-Newton method (secant method) aims at finding a matrix $\mathbf{B}^{(k)}$ which satisfies

$$\mathbf{B}^{(k)}\mathbf{s}^{(k)} = \mathbf{y}^{(k)} \tag{25}$$

in order to approximate $\mathbf{H}^{(k)}$. Here, $\mathbf{s}^{(k)} = \mathbf{w}^{(k)} - \mathbf{w}^{(k-1)}$ and $\mathbf{y}^{(k)} = \nabla_{\mathbf{w}^*}E(\mathbf{w}^{(k)}) - \nabla_{\mathbf{w}^*}E(\mathbf{w}^{(k-1)})$. In order to obtain the optimal stepsize for (7), we use $\alpha_k\mathbf{I}$ to replace $\mathbf{B}^{(k)}$ in (25), where α_k is a scalar. Obviously, in general, we cannot find $\alpha_k\mathbf{I}$ that satisfies (25). Instead, we compute α_k by solving the least squares problem

$$\begin{aligned}\alpha_k &= \arg\min_{\alpha\in\mathbb{C}} \|\alpha\mathbf{I}\mathbf{s}^{(k)} - \mathbf{y}^{(k)}\|^2\\ &= \arg\min_{\alpha\in\mathbb{C}} (\alpha\mathbf{s}^{(k)} - \mathbf{y}^{(k)})^H(\alpha\mathbf{s}^{(k)} - \mathbf{y}^{(k)})\end{aligned} \tag{26}$$

to give

$$\alpha_k = \frac{(\mathbf{s}^{(k)})^H\mathbf{y}^{(k)}}{(\mathbf{s}^{(k)})^H\mathbf{s}^{(k)}}. \tag{27}$$

Accordingly, the optimal stepsize η_k at the iteration k can be obtained as

$$\eta_k = \frac{1}{\alpha_k} = \frac{(\mathbf{s}^{(k)})^H \mathbf{s}^{(k)}}{(\mathbf{s}^{(k)})^H \mathbf{y}^{(k)}}. \tag{28}$$

By analogy, we can set

$$\begin{aligned} \beta_k &= \arg\min_{\beta \in \mathbb{C}} \|\mathbf{s}^{(k)} - \beta \mathbf{y}^{(k)}\|^2 \\ &= \arg\min_{\beta \in \mathbb{C}} (\mathbf{s}^{(k)} - \beta \mathbf{y}^{(k)})^H (\mathbf{s}^{(k)} - \beta \mathbf{y}^{(k)}) \end{aligned} \tag{29}$$

to give

$$\eta_k = \beta_k = \frac{(\mathbf{y}^{(k)})^H \mathbf{s}^{(k)}}{(\mathbf{y}^{(k)})^H \mathbf{y}^{(k)}} \tag{30}$$

as another choice of the optimal stepsize. We should mention that the stepsize η_k given by (28) and that given by (30) are both complex-valued. Moreover, as

$$\|(\mathbf{y}^{(k)})^H \mathbf{s}^{(k)}\| \le \|(\mathbf{y}^{(k)})\| \|\mathbf{s}^{(k)}\| \tag{31}$$

we have

$$\left| \frac{(\mathbf{s}^{(k)})^H \mathbf{s}^{(k)}}{(\mathbf{s}^{(k)})^H \mathbf{y}^{(k)}} \right| \ge \left| \frac{(\mathbf{y}^{(k)})^H \mathbf{s}^{(k)}}{(\mathbf{y}^{(k)})^H \mathbf{y}^{(k)}} \right|. \tag{32}$$

Thus, a complex gradient method with the complex stepsize given by (30) will exhibit enhanced stability compared with its counterpart given by (28), and we shall, therefore, mainly discuss the case in (30) in the rest of this brief.

Notice that we can also solve the least squares problem in (29) in the real domain as follows:

$$\begin{aligned} \beta_k^R &= \arg\min_{\beta \in \mathbb{R}} \|\mathbf{s}^{(k)} - \beta \mathbf{y}^{(k)}\|^2 \\ &= \arg\min_{\beta \in \mathbb{R}} (\mathbf{s}^{(k)} - \beta \mathbf{y}^{(k)})^H (\mathbf{s}^{(k)} - \beta \mathbf{y}^{(k)}). \end{aligned} \tag{33}$$

The so-obtained real stepsize is given by

$$\eta_k^R = \beta_k^R = \frac{\Re((\mathbf{y}^{(k)})^H \mathbf{s}^{(k)})}{(\mathbf{y}^{(k)})^H \mathbf{y}^{(k)}} \tag{34}$$

for which

$$\|\mathbf{s}^{(k)} - \beta_k \mathbf{y}^{(k)}\|^2 \le \|\mathbf{s}^{(k)} - \beta_k^R \mathbf{y}^{(k)}\|^2. \tag{35}$$

Recall that the basic idea behind the BBM is to find β_k by minimizing $\|\mathbf{s}^{(k)} - \beta \mathbf{y}^{(k)}\|^2$, such that $\beta_k \mathbf{I}$ provides a rational approximation for the inverse of the Hessian matrix $\mathbf{H}^{(k)}$. In this sense, (35) implies that the complex stepsize η_k, defined by (30), is a better choice than its real counterpart η_k^R.

V. Simulation Results

We verified the theoretical analysis on a complex-valued approximation problem. The function to be approximated is given by [32]

$$f(\mathbf{z}) = \frac{1}{1.5} \left(\frac{z_2^2}{z_1} + z_3 + 10 z_1 z_4 \right) \tag{36}$$

where $\mathbf{z} = (z_1, z_2, z_3, z_4)^T \in \mathbb{C}^4$. Training samples were generated by randomly choosing the real and imaginary parts of $z_l (l = 1, 2, 3, 4)$ from the uniform distribution in the range $[-0.5, 0.5]$.

We used a single hidden-layer CVNN with the structure $4-50-1$ to approximate 50 points of the function (36). The activation functions of the neurons were $\tanh(\cdot)$ functions. The network was trained using the following five methods: 1) CGLM with a constant real stepsize $\eta = 0.02$; 2) CGLM with a constant complex stepsize $\eta = 0.02 - 0.01i$; 3) CGLM with a constant pure imaginary stepsize $\eta = 0.02i$; 4) CBBM-CSS from (30); and 5) CBBM-RSS from (34). For convenience, in every simulation trial, the methods shared the same initial weights (both the real part and the imaginary part) that were taken as random numbers from the interval $[-0.2, 0.2]$.

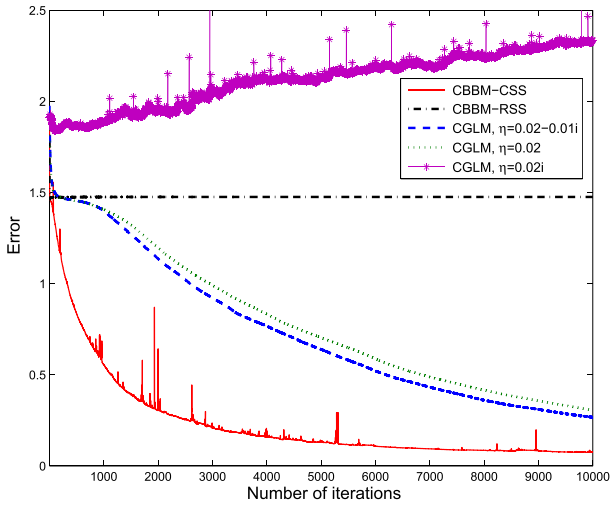

Fig. 3. Learning curves for CGLM with different stepsizes.

The average performance curves based on averaging 50 independent trials are shown in Fig. 3, and indicate the following.

1) Both the CGLMs with constant stepsizes $\eta = 0.02$ and $\eta = 0.02 - 0.01i$ were converged. Due to the enhanced ability to escape the saddle points and the increase in the modulus, the constant complex stepsize outperformed the corresponding constant real stepsize in terms of convergence speed. The CGLM with the pure imaginary stepsize $\eta = 0.02i$ failed to converge.
2) By adaptively choosing the stepsize in the complex domain, the CBBM-CSS converged almost six times faster than the conventional CGLM, whereas the CBBM-RSS almost failed to learn the data. This exemplifies that the CBBM-CSS, but not the CBBM-RSS, is a generic extension of the original BBM to the complex domain.

VI. Conclusion

We have introduced a CGLM with a complex-valued stepsize, which has demonstrated significant advantages in CGLMs. Our findings can be summarized as follows.

1) Compared with the traditional real stepsize, the complex stepsize extends the search space of CGLM from a half-line to a half-plane and reduces the risk of getting stuck into the plateaus around the saddle points.
2) To guarantee the convergence of CGLM with a complex stepsize, the real part of the stepsize should be a positive number, while there is no sign restriction on the imaginary part. Pure imaginary numbers cannot serve as the stepsize.
3) During the learning process, if the iteration is near a local minimum of the error function, a constant complex stepsize should be avoided. Constant complex stepsizes are also not recommended for linear networks.

Furthermore, we have derived two CBBM-CSS and CBBM-RSS. Both the theoretical and experimental analyses have shown the superiority of the complex stepsize.

REFERENCES

[1] D. P. Mandic and V. S. L. Goh, *Complex Valued Nonlinear Adaptive Filters: Noncircularity, Widely Linear and Neural Models*. New York, NY, USA: Wiley, 2009.

[2] I. N. Aizenberg, N. N. Aizenberg, and J. P. L. Vandewalle, *Multi-Valued and Universal Binary Neurons: Theory, Learning and Applications*. Norwell, MA, USA: Kluwer, 2000.

[3] A. Hirose, *Complex-Valued Neural Networks*. New York, NY, USA: Springer-Verlag, 2006.

[4] T. Xiong, Y. Bao, Z. Hu, and R. Chiong, "Forecasting interval time series using a fully complex-valued RBF neural network with DPSO and PSO algorithms," *Inf. Sci.*, vol. 305, pp. 77–92, Jun. 2015.

[5] T. Nitta, "Orthogonality of decision boundaries in complex-valued neural networks," *Neural Comput.*, vol. 16, no. 1, pp. 73–97, 2004.

[6] C. Jahanchahi, S. Kanna, and D. P. Mandic, "Complex dual channel estimation: Cost effective widely linear adaptive filtering," *Signal Process.*, vol. 104, pp. 33–42, Nov. 2014.

[7] S. C. Douglas, "Fixed-point algorithms for the blind separation of arbitrary complex-valued non-Gaussian signal mixtures," *EURASIP J. Appl. Signal Process.*, vol. 2007, pp. 1–15, Jan. 2007.

[8] T. Nitta, "An extension of the back-propagation algorithm to complex numbers," *Neural Netw.*, vol. 10, no. 8, pp. 1391–1415, 1997.

[9] A. Hirose, "Continuous complex-valued back-propagation learning," *Electron. Lett.*, vol. 28, no. 20, pp. 1854–1855, 1992.

[10] T. Kim and T. Adali, "Fully complex multi-layer perceptron network for nonlinear signal processing," *J. VLSI Signal Process. Syst. Signal, Image Video Technol.*, vol. 32, nos. 1–2, pp. 29–43, 2002.

[11] T. Kim and T. Adali, "Approximation by fully complex multilayer perceptrons," *Neural Comput.*, vol. 15, no. 7, pp. 1641–1666, Jul. 2003.

[12] S. L. Goh and D. P. Mandic, "A complex-valued RTRL algorithm for recurrent neural networks," *Neural Comput.*, vol. 16, no. 12, pp. 2699–2713, Dec. 2004.

[13] W. Wirtinger, "Zur formalen theorie der funktionen von mehr komplexen veränderlichen," *Math. Ann.*, vol. 97, no. 1, pp. 357–375, Dec. 1927.

[14] D. H. Brandwood, "A complex gradient operator and its application in adaptive array theory," *IEE Proc. F Commun., Radar Signal Process.*, vol. 130, no. 1, pp. 11–16, Feb. 1983.

[15] K. Kreutz-Delgado, "The complex gradient operator and the $\mathbb{CR}$ calculus," Dept. Elect. Comput. Eng., Univ. California, San Diego, CA, USA, Tech. Rep. ECE275A, 2006.

[16] H. Li and T. Adali, "Complex-valued adaptive signal processing using nonlinear functions," *EURASIP J. Adv. Signal Process.*, vol. 2008, pp. 1–9, Feb. 2008, Art. ID 765615.

[17] S. Javidi, D. P. Mandic, and A. Cichocki, "Complex blind source extraction from noisy mixtures using second-order statistics," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 7, pp. 1404–1416, Jul. 2010.

[18] R. Savitha, S. Suresh, and N. Sundararajan, "Projection-based fast learning fully complex-valued relaxation neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 4, pp. 529–541, Apr. 2013.

[19] D. P. Xu, H. S. Zhang, and D. P. Mandic, "Convergence analysis of complex CR-gradient algorithms for complex-valued neural networks with non-analytic activation functions," submitted for publication.

[20] D. P. Mandic, "A generalized normalized gradient descent algorithm," *IEEE Signal Process. Lett.*, vol. 11, no. 2, pp. 115–118, Feb. 2004.

[21] S. L. Goh and D. P. Mandic, "Stochastic gradient-adaptive complex-valued nonlinear neural adaptive filters with a gradient-adaptive step size," *IEEE Trans. Neural Netw.*, vol. 18, no. 5, pp. 1511–1516, Sep. 2007.

[22] M. F. Amin, M. I. Amin, A. Y. H. Al-Nuaimi, and K. Murase, "Wirtinger calculus based gradient descent and Levenberg–Marquardt learning algorithms in complex-valued neural networks," in *Neural Information Processing*, vol. 7062, B.-L. Lu, L. Zhang, and J. Kwok, Eds. Berlin, Germany: Springer-Verlag, 2011, pp. 550–559.

[23] Y. Chen, D. Han, and L. Qi, "New ALS methods with extrapolating search directions and optimal step size for complex-valued tensor decompositions," *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 5888–5898, Dec. 2011.

[24] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA J. Numer. Anal.*, vol. 8, no. 1, pp. 141–148, 1988.

[25] M. Raydan, "The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem," *SIAM J. Optim.*, vol. 7, no. 1, pp. 26–33, 1997.

[26] T. Nitta, "Local minima in hierarchical structures of complex-valued neural networks," *Neural Netw.*, vol. 43, pp. 1–7, Jul. 2013.

[27] K. Fukumizu and S. Amari, "Local minima and plateaus in hierarchical structures of multilayer perceptrons," *Neural Netw.*, vol. 13, no. 3, pp. 317–327, 2000.

[28] H. Zhang, X. Liu, D. Xu, and Y. Zhang, "Convergence analysis of fully complex backpropagation algorithm based on Wirtinger calculus," *Cognit. Neurodyn.*, vol. 8, no. 3, pp. 261–266, Jun. 2014.

[29] D. Xu, H. Zhang, and D. P. Mandic, "Convergence analysis of an augmented algorithm for fully complex-valued neural networks," *Neural Netw.*, vol. 69, pp. 44–50, Sep. 2015.

[30] H. Zhang, D. Xu, and Y. Zhang, "Boundedness and convergence of split-complex back-propagation algorithm with momentum and penalty," *Neural Process. Lett.*, vol. 39, no. 3, pp. 297–307, Jun. 2014.

[31] M. V. Konnik and J. De Doná, "Feasibility of constrained receding horizon control implementation in adaptive optics," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 1, pp. 274–289, Jan. 2015.

[32] K. Subramanian, R. Savitha, and S. Suresh, "A complex-valued neuro-fuzzy inference system and its learning mechanism," *Neurocomputing*, vol. 123, pp. 110–120, Jan. 2014.