Chapter 4
Fully Complex-valued Relaxation Networks

The complex-valued learning algorithms described in the chapters 2 and 3 use a real-valued mean square error function as the performance measure which explicitly minimizes only the magnitude error. In addition, the mean squared error function is non-analytic in the Complex domain (not differentiable in an open set). Therefore, pseudo-gradients, or isomorphic $\mathbb{C}^1 \rightarrow \Re^2$ transformations are generally used in the derivation of the learning algorithms. Use of a mean squared error function which is only an explicit representation of the magnitude error and the the use of pseudo-gradients will affect the phase approximation capabilities of these algorithms. For better phase approximation, one needs to use an error function which simultaneously minimizes both the magnitude and phase errors [1]. But if such functions are inseparable into their real and imaginary parts, then the pseudo gradients are invalid and one needs to identify a mathematical tool to derive the gradients of such functions. Hence, there is a need for the development of a fully complex-valued neural network and its learning algorithm using the fully complex-valued gradients to overcome the above-mentioned issues.

In this chapter, we present a fully complex-valued single hidden layer neural network with a Gaussian like activation function in the hidden layer and an exponential activation function in the output layer. For a given training data set and number of hidden neurons, the network parameters are analytically estimated using a projection based learning algorithm. The learning algorithm employs a nonlinear logarithmic error function as the energy function which explicitly contains both the magnitude and phase errors. The problem of finding the optimal weights is formulated as a nonlinear programming problem. The problem is solved with the help of Wirtinger calculus [2]. Wirtinger calculus provides a framework for determining gradients of a non-analytic nonlinear complex (energy) function. The projection based learning algorithm converts the nonlinear programming problem into solving a system of linear equations and provides a solution for the optimal weights corresponding to the minimum energy point of the energy function. This is similar to the relaxation process, where the system always returns to a minimum energy state from a given initial condition [3]. Therefore, we refer to the proposed complex-valued network as a, ‘Fully Complex-valued Relaxation Network (FCRN)’. The projection based

S. Suresh et al.: Supervised Learning with Complex-valued Neural Networks, SCI 421, pp. 73–83.
springerlink.com

learning algorithm of FCRN requires minimal computational effort to approximate any desired function with higher accuracy. In the next sections, we present FCRN and its learning algorithm in detail.

4.1 Fully Complex-valued Relaxation Networks

Complex-valued neural networks are generally employed to handle applications where the signals involved are inherently complex-valued. An efficient complex-valued neural network is required to preserve the nonlinear transformations (both in magnitude and phase) between the complex-valued inputs and their corresponding targets with a minimal computational effort. In this section, we present one such complex-valued neural network and its "projection based learning" algorithm. For a given training data set and the number of hidden neurons, the network parameters are estimated as a solution to a nonlinear programming problem using Wirtinger calculus. The projection based learning algorithm converts the nonlinear programming problem into a system of linear equations and the solution of the same results in computing the optimal output weights. The system of linear equations is derived from a nonlinear logarithmic energy function that contains both the magnitude and phase errors explicitly and the optimal output weights are obtained corresponding to the minimum energy point of this energy function. This process is analogous to a relaxation process, where a system always returns to the minimum energy state from any given initial condition. Hence, the proposed complex-valued neural network is referred to as a, 'Fully Complex-valued Relaxation Network (FCRN)'. The architecture and the learning algorithm of FCRN is described in detail below.

4.1.1 FCRN Architecture

The fully complex-valued relaxation network is a single hidden layer complex-valued neural network having a linear input layer with m neurons, a non-linear hidden layer with h neurons and a non-linear output layer with n neurons, as shown in Fig. 4.1. The neurons in the hidden layer employ a hyperbolic secant function (*sech*) whose magnitude response is similar to that of the real-valued Gaussian activation function.

For a given m-dimensional input $\mathbf{z}^t = [z_1^t, \cdots, z_m^t]^T \in \mathbb{C}^m$, the response of the k-th hidden neuron ($z_h^{k^t}$) is given by:

$$z_h^{k^t} = sech\left(\boldsymbol{\sigma}_k^T \left(\mathbf{z}^t - \mathbf{c}_k\right)\right);\ k = 1, \cdots, h \tag{4.1}$$

where $\boldsymbol{\sigma}_k = [\sigma_k^1, \cdots, \sigma_k^m]^T \in \mathbb{C}^m$ is the scaling factor of the k-th hidden neuron, $\mathbf{c}_k = [c_k^1, \cdots, c_k^m]^T \in \mathbb{C}^m$ is the center of the k-th hidden neuron, the superscript T represents the transpose operator, and $sech(z) = \frac{2}{e^z + e^{-z}}$.

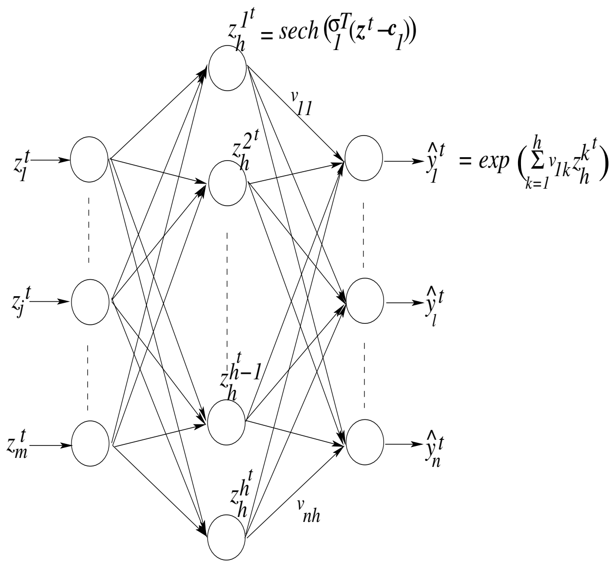

Fig. 4.1 The architecture of FCRN

The n-dimensional output of the network is given by $\widehat{\mathbf{y}}^t = [\widehat{y}_1^t, \cdots, \widehat{y}_n^t]^T \in \mathbb{C}^n$. The neurons in the output layer employ an *exponential* activation function and the response of the l-th output neuron is:

$$\widehat{y}_l^t = exp\left(\sum_{k=1}^{K} v_{lk} z_h^{k^t}\right); \; l = 1, \cdots, n \tag{4.2}$$

where $v_{lk} \in \mathbb{C}$ is the weight connecting the k-th hidden neuron and the l-th output neuron.

Given a training data set $\{(\mathbf{z}^1, \mathbf{y}^1), \cdots, (\mathbf{z}^t, \mathbf{y}^t), \cdots, (\mathbf{z}^N, \mathbf{y}^N)\}$, with $\mathbf{z}^t \in \mathbb{C}^m$ is the m-dimensional input and $\mathbf{y}^t \in \mathbb{C}^n$ is the n-dimensional target of the t-th training sample. The main objective of the fully complex-valued relaxation network is to estimate the free parameters of the network ($\mathbf{C}$, $\boldsymbol{\sigma}$ and $\mathbf{V}$) such that the predicted output ($\widehat{\mathbf{y}}$) is as close as possible to the target output ($\mathbf{y}$), with a given number of hidden neurons (h). In other words, FCRN is required to approximate the underlying transformation function ($\mathbf{F} : \mathbf{z}^t \rightarrow \mathbf{y}^t$) as accurately as possible. Most of the learning algorithms reported in the literature use the mean square error deviation between actual ($\mathbf{y}^t$) and predicted output ($\widehat{\mathbf{y}}^t$) as the performance criterion, which is only an explicit minimization of the magnitude of error. However, accurate estimation of both magnitude and phase of the signals are important in many real-world applications involving complex-valued signals [4]. In this chapter, we use a nonlinear logarithmic error function as the energy function with an explicit representation of both the magnitude and phase of the actual and predicted outputs.

4.1.2 Nonlinear Logarithmic Energy Function

The actual output ($\mathbf{y}^t$) of the t-th training sample is represented in polar form as

$$y_l^t = r_l^t \, exp\left(i\phi_l^t\right); \; l = 1, 2, \cdots, n \tag{4.3}$$

where $r_l^t = ||y_l^t||$ is the magnitude of y_l^t and $\phi_l^t = arctan\left(\frac{Im(y_l^t)}{Re(y_l^t)}\right)$[1] is the phase of y_l^t.[2]

Similarly, the predicted output ($\widehat{\mathbf{y}}^t$) of t-th training sample is represented in polar form as

$$\widehat{y}_l^t = \widehat{r}_l^t \, exp\left(i\widehat{\phi}_l^t\right); \; l = 1, 2, \cdots, n \tag{4.4}$$

where $\widehat{r}_l^t = ||\widehat{y}_l^t||$ is the estimated magnitude and $\widehat{\phi}_l^t = arctan\left(\frac{Im(\widehat{y}_l^t)}{Re(\widehat{y}_l^t)}\right)$ is the estimated phase.

The energy function J_t should be a monotonically decreasing function that represents the magnitude and phase quantities explicitly, i.e., $J_t \to 0$ when $\widehat{\mathbf{r}}^t \to \mathbf{r}^t$ and $\widehat{\boldsymbol{\phi}}^t \to \boldsymbol{\phi}^t$.

We propose an energy function that uses a logarithmic function for explicit representation of both the magnitude and phase of complex signals and is of the form

$$J_t = \sum_{l=1}^{n} \left(ln(\widehat{y}_l^t) - ln(y_l^t)\right) \overline{\left(ln(\widehat{y}_l^t) - ln(y_l^t)\right)} \tag{4.5}$$

where $\overline{(.)}$ is the conjugate of the complex signal $(.)$ and $ln(.)$ represents the natural logarithmic function.

Substituting the polar representation of actual (y_l^t) and predicted output ($\widehat{y}_l^t$), the above equation reduces to

$$J_t = \sum_{l=1}^{n} \left(ln\left(\frac{\widehat{r}_l^t}{r_l^t}\right)^2 + \left(\widehat{\phi}_l^t - \phi_l^t\right)^2 \right) \tag{4.6}$$

It can be observed from Eq. (4.6) that the logarithmic energy function represents the magnitude and phase quantities explicitly and J_t tends to 0, when $\widehat{y}_l^t \to y_l^t$. It must also be noted that the energy function is second order continuously differentiable with respect to the network parameters.

For N training samples, the overall energy is defined as

$$\begin{aligned} J(\mathbf{V}) &= \frac{1}{2}\sum_{t=1}^{N} J_t \\ &= \frac{1}{2}\sum_{t=1}^{N}\sum_{l=1}^{n} \left(ln\left(\frac{\widehat{r}_l^t}{r_l^t}\right)^2 + \left(\widehat{\phi}_l^t - \phi_l^t\right)^2 \right) \end{aligned} \tag{4.7}$$

[1] $y_l^t = Re(y_l^t) + i \, Im(y_l^t)$, where $Re(.)$ and $Im(.)$ refers to the real and imaginary parts of a complex number, respectively

[2] Note that $i = \sqrt{-1}$ is the Complex operator.

In the next section, we derive the projection based learning algorithm of FCRN such that $J(\mathbf{W})$ is minimum.

4.1.3 A Projection Based Learning Algorithm for FCRN

For a given initial condition (i.e., N training samples, h hidden neurons), the projection based learning algorithm finds the network parameters for which the energy function is minimum, i.e., the network achieves the minimum energy point or relaxation point of the energy function.

The hidden neuron centers ($\mathbf{c}_k$) and scaling factors ($\boldsymbol{\sigma}_k$) of FCRN are chosen as random constants and the optimal output weights ($\mathbf{V}^* \in \mathbb{C}^{n\times h}$) are estimated such that the total energy reaches its minimum.

$$\mathbf{V}^* := arg \min_{\mathbf{V}\in\mathbb{C}^{n\times h}} J(\mathbf{V}) \tag{4.8}$$

The problem of estimating the optimal weight is converted to an unconstrained minimization problem ($J(\mathbf{V}) : \mathbb{C}^{n\times h} \to \Re$) involving minimization of the energy function $J(\mathbf{V})$. Let $\mathbf{V}^* \in \mathbb{C}^{n\times h}$, then $\mathbf{V}^*$ is the optimal output weight corresponding to the minimum of the energy function if $J(\mathbf{V}^*) \leq J(\mathbf{V})\ \forall\ \mathbf{V} \in \mathbb{C}^{n\times h}$. The optimal $\mathbf{V}^*$ corresponding to the minimum energy point of the energy function ($J(\mathbf{V}^*)$) is obtained by equating the first order partial derivative of $J(\mathbf{V})$ with respect to the output weight to zero, i.e.,

$$\frac{\partial J(\mathbf{V})}{\partial v_{lp}} = 0;\ \ l = 1,\cdots,n;\ p = 1,\cdots,h \tag{4.9}$$

For convenience, we rewrite the energy function as

$$J(\mathbf{V}) = \frac{1}{2}\sum_{t=1}^{N}\sum_{l=1}^{n}(ln(\widehat{y}_l^t)-ln(y_l^t))\,\overline{(ln(\widehat{y}_l^t)-ln(y_l^t))} \tag{4.10}$$

By substituting the predicted output ($\widehat{\mathbf{y}}^t$) from Eq. (8.17) in Eq. (4.10), the energy function reduces to

$$J(\mathbf{V}) = \frac{1}{2}\sum_{t=1}^{N}\sum_{l=1}^{n}\left(\sum_{k=1}^{K} v_{lk}z_h^{kt} - ln\ y_l^t\right)\overline{\left(\sum_{k=1}^{K} v_{lk}z_h^{kt} - ln\ y_l^t\right)} \tag{4.11}$$

where z_h^{kt} is the response of the k-th hidden neuron for t-th training sample.

Since the energy function is a non-analytic, non-linear real-valued function of the complex-valued output weights and is inseparable into its real and imaginary parts, we use Wirtinger calculus[1] [2] to obtain the first order partial derivatives of the energy function with respect to the complex-valued output weight (v_{lp}). Wirtinger calculus eliminates the stringent conditions of analyticity for the *Complex differentiability* imposed by the Cauchy Riemann conditions. They define the Complex differentiability of almost all functions of interest, including the energy function that maps ($\mathbb{C} \to \Re$). Although the derivatives defined by Wirtinger calculus do not satisfy the Cauchy Riemann equations, they obey all the rules of calculus (like differentiation of products, chain rule etc.).

Using the Wirtinger calculus and the commutative property of the Complex conjugate operator[2], the first order partial derivative of energy function with respect to w_{lp} ($l = 1, 2 \cdots, n$ and $p = 1, 2, \cdots, h$) is given as:

$$\frac{\partial J(\mathbf{W})}{\partial v_{lp}} = \sum_{t=1}^{N} z_h^{p^t} \left[\sum_{k=1}^{h} \overline{v}_{lk} \overline{z}_h^{k^t} - ln\left(\overline{y}_l^t\right) \right] \tag{4.15}$$

Equating the first partial derivative to zero and re-arranging the Eq. (4.15), we get

$$\sum_{k=1}^{h} \overline{v}_{lk} \sum_{t=1}^{N} z_h^{p^t} \overline{z_h}^{k^t} = \sum_{t=1}^{N} ln\left(\overline{y}_l^t\right) z_h^{p^t} \tag{4.16}$$

Eq. (4.16) can be written as

$$\sum_{k=1}^{h} \overline{v}_{lk} A_{pk} = B_{lp};\ p = 1, \cdots, h;\ l = 1, \cdots, n \tag{4.17}$$

which can be represented in matrix form as

[1] Let $f_R(z_c, \overline{z}_c)$ be a real-valued function of a complex-valued variable $z_c = x_r + iy_r$. Then, the following pair of derivatives are defined by the Wirtinger calculus:

$$\mathbb{R}\text{-derivative of } f_R(z_c, \overline{z}_c) = \frac{\partial f_R}{\partial z_c} |_{\overline{z_c}=\text{constant}} \tag{4.12}$$

$$\overline{\mathbb{R}}\text{-derivative of } f_R(z, \overline{z}_c) = \frac{\partial f_R}{\partial \overline{z_c}} |_{z_c=\text{constant}} \tag{4.13}$$

It is proved in [5] that the $\mathbb{R}-derivative$ (Eq. (4.12)) and the $\overline{\mathbb{R}}-derivative$ (Eq. (4.13)) can be equivalently written as

$$\frac{\partial f_R}{\partial z_c} = \frac{1}{2}\left(\frac{\partial f_R}{\partial x_r} - i\frac{\partial f_R}{\partial y_r}\right)$$
$$\frac{\partial f_R}{\partial \overline{z}_c} = \frac{1}{2}\left(\frac{\partial f_R}{\partial x_r} + i\frac{\partial f_R}{\partial y_r}\right) \tag{4.14}$$

where the partial derivatives with respect to x_r and y_r are *true* partial derivatives of the function $f_R(z_c) = f_R(x_r, y_r)$, which is differentiable with respect to the x_r and y_r.

[2] $\overline{z_a + z_b} = \overline{z_a} + \overline{z_b}$ and $\overline{ln(z_a)} = ln(\overline{z_a})$

$$\overline{\mathbf{V}}\mathbf{A} = \mathbf{B} \tag{4.18}$$

where the projection matrix $\mathbf{A} \in \mathbb{C}^{h\times h}$ is given by

$$A_{pk} = \sum_{t=1}^{N} z_h^{p^t} \overline{z_h^{k^t}};\ p = 1,\cdots,h;\ k = 1,\cdots,h \tag{4.19}$$

and the output matrix $\mathbf{B} \in \mathbb{C}^{n\times h}$ is

$$B_{lp} = \sum_{t=1}^{N} ln\ \bar{y}_l^t z_h^{k^t};\ l = 1,\cdots,n;\ p = 1,\cdots,h \tag{4.20}$$

Eq. (4.17) gives the set of $n \times h$ linear equations with $n \times h$ unknown output weights $\mathbf{V}$. Note that the projection matrix is always a square matrix of order $h \times h$.

We state the following propositions to find the closed-form solution for these set of linear equations.

Proposition 4.1. *The responses of the neurons in the hidden layer are unique. i.e. $\forall\ \mathbf{z}^t$, when $k \neq p$, $z_h^{k^t} \neq z_h^{p^t}$; $k,p = 1,2\cdots h$, $t = 1,\cdots h$.*

Proof. Let us assume that

$$\text{For a given } \mathbf{z}^t, z_h^{p^t} = z_h^{k^t};\ k \neq p \tag{4.21}$$

This assumption is valid if and only if

$$\begin{aligned} sech\left(\boldsymbol{\sigma}_p^T(\mathbf{z}^t - \mathbf{c}_p)\right) &= sech\left(\boldsymbol{\sigma}_k^T(\mathbf{z}^t - \mathbf{c}_k)\right) \\ \text{OR}\ \ \boldsymbol{\sigma}_p^T(\mathbf{z}^t - \mathbf{c}_p) &= \boldsymbol{\sigma}_k^T(\mathbf{z}^t - \mathbf{c}_k) \end{aligned} \tag{4.22}$$

The pair of parameters c_{kj} and c_{pj} (that are elements of the vectors $\mathbf{c}_k$ and $\mathbf{c}_p$, respectively), σ_{kj} and σ_{pj} (that are elements of the vectors $\boldsymbol{\sigma}_k$ and $\boldsymbol{\sigma}_p$, respectively) are uncorrelated random constants chosen from a ball of radius 1, i.e.,

$$||c_{kj}|| < 1; ||\sigma_{kj}|| < 1;\ k = 1,\cdots,h;\ j = 1,\cdots,m \tag{4.23}$$

Therefore, $\mathbf{c}_k \neq \mathbf{c}_p$ and $\boldsymbol{\sigma}_k \neq \boldsymbol{\sigma}_p$ for any $\mathbf{z}^t$ (the t-th random input vector of the training data with N samples). Hence, the response of the k-th and p-th hidden neurons are not equal, i.e., $z_h^{p^t} \neq z_h^{k^t}\ \forall\ \mathbf{z}^t;\ t = 1,\cdots,N$.

Proposition 4.2. *The responses of the neurons in the hidden layer are non-zero. i.e $\forall\ \mathbf{z}$, $z_h^{k^t} \neq 0$; $k = 1,2\cdots h$.*

Proof. Let us assume that the hidden layer response of the k-th hidden neuron is 0, i.e.,

$$z_h^{k^t} = 0 \tag{4.24}$$

This is possible if and only if

$$\begin{aligned} \mathbf{c}_k^T(\mathbf{z}^t - \boldsymbol{\sigma}_k) &= \infty \\ \mathbf{z} \to \infty, \text{ or } \quad \mathbf{c}_k &\to \infty, \text{ or } \boldsymbol{\sigma}_k \to \infty \end{aligned} \tag{4.25}$$

As stated in Eq. (4.23), the hidden layer parameters are random constants chosen from within a circle of radius 1. The input variables $\mathbf{z}^t$ are also normalized in a circle of radius 1 such that

$$|z_j| < 1; \; j = 1, \cdots, m \tag{4.26}$$

Hence, the assumption in Eq. (4.24) is not valid for all $\mathbf{z}^t$. Thus, the responses of the neurons in the hidden layer $z_h^{k^t} \neq 0 \, \forall \, \mathbf{z}^t$.

Using the *Proposition* 4.1 and *Proposition* 4.2, we state the following theorem.

Theorem 4.1. *The projection matrix* **A** *is a positive definite Hermitian matrix, and hence, it is invertible.*

Proof. From the definition of the projection matrix **A** given in Eq. (4.19),

$$\mathbf{A}_{pk} = \sum_{t=1}^{N} z_h^{p^t} \overline{z_h}^{k^t}; \; p = 1, \cdots, h; \; k = 1, \cdots, h \tag{4.27}$$

it can be derived that the diagonal elements of the **A** for the t-th sample is:

$$A_{kk}^t = z_h^{k^t} \overline{z_h}^{k^t}; \; k = 1, \cdots, h \tag{4.28}$$

From *Proposition* 4.2, the responses of the hidden neurons are non-zero. Hence, $A_{kk}^t \neq 0$. Therefore Eq. (4.28) can be written as

$$A_{kk}^t = |z_h^{k^t}|^2 > 0 \tag{4.29}$$

Hence the diagonal elements of the projection matrix are real, and positive, i.e., $A_{kk}^t \in \Re > 0$. This can be extended for the entire training sample set as:

$$\mathbf{A}_{kk} = \sum_{t=1}^{N} A_{kk}^t \in \Re > 0 \tag{4.30}$$

The off-diagonal elements of the projection matrix (**A**) for the t-th sample is:

$$A_{kj}^t = z_h^{k^t} \overline{z_h}^{j^t} \text{ and } A_{jk}^t = z_h^{j^t} \overline{z_h}^{k^t} \tag{4.31}$$

$$\Longrightarrow A_{kj}^t = \overline{A}_{jk}^t \tag{4.32}$$

Using the commutative property of the complex conjugate operator, Eq. (4.31) can be extended for all the N samples as:

$$\mathbf{A}_{kj} = \sum_{t=1}^{N} A^t_{kj} = \sum_{t=1}^{N} \overline{A}_{jk} = \overline{\mathbf{A}}_{jk} \tag{4.33}$$

From Eqs. 4.30 and 4.33, it can be inferred that the projection matrix $\mathbf{A}$ is a Hermitian matrix.

A Hermitian matrix is positive definite *iff* for any $\mathbf{q} \neq 0$, $\mathbf{q}^H\mathbf{A}\mathbf{q} > 0$. Let us consider an unit basis vector $\mathbf{q}_1 \in \Re^{K\times 1}$ such that $q_{11} = 1$ and $q_{12}\cdots q_{1K} = 0$, i.e., $\mathbf{q}_1 = [1\cdots 0\cdots 0\cdots 0]^T$. Therefore,

$$\mathbf{q}_1^H\mathbf{A}\mathbf{q}_1 = \mathbf{A}_{11} \tag{4.34}$$

In Eq. (4.30), it was shown that $k = 1,\cdots,h, \mathbf{A}_{kk} \in \Re > 0$. Therefore,

$$\mathbf{A}_{11} \in \Re > 0 \implies \mathbf{q}_1^H\mathbf{A}\mathbf{q}_1 > 0 \tag{4.35}$$

Similarly, for an unit basis vector $\mathbf{q}_k = [0\cdots 1\cdots 0]^T$, the product $\mathbf{q}_k^H\mathbf{A}\mathbf{q}_k$ is given by

$$\mathbf{q}_k^H\mathbf{A}\mathbf{q}_k = \mathbf{A}_{kk} > 0;\ k = 1,\cdots,h \tag{4.36}$$

Let $\mathbf{p} \in \mathbb{C}^h$ be the linear transformed sum of h such unit basis vectors, i.e., $\mathbf{p} = \mathbf{q}_1 t_1 + \cdots + \mathbf{q}_k t_k + \cdots + \mathbf{q}_h t_h$, where $t_k \in \mathbf{C}$ is the transformation constant. Then,

$$\begin{aligned}
\mathbf{p}^H\mathbf{A}\mathbf{p} &= \sum_{k=1}^{h} (\mathbf{q}_k t_k)^H \mathbf{A} \sum_{k=1}^{h} (\mathbf{q}_k t_k) \\
&= \sum_{k=1}^{h} |t_k|^2 \left(\mathbf{q}_k^H\mathbf{A}\mathbf{q}_k\right) \\
&= \sum_{k=1}^{h} |t_k|^2 \mathbf{A}_{kk}
\end{aligned} \tag{4.37}$$

As shown in Eq. (4.30), $\mathbf{A}_{kk} \in \Re > 0$. Also, that $|t_k|^2 \in \Re > 0$ is evident. Hence,

$$\begin{aligned}
&|t_k|^2\mathbf{A}_{kk} \in \Re > 0\ \forall\ k \\
\implies &\sum_{k=1}^{h} |t_k|^2\mathbf{A}_{kk} \in \Re > 0
\end{aligned} \tag{4.38}$$

Thus, the projection matrix A is positive definite, and is hence, invertible.

The solution for $\mathbf{V}$ obtained as a solution to the set of equations, given in Eq. (4.18) is minimum, if $\frac{\partial^2 J}{\partial {v_{lp}}^2} > 0$. The second derivative of the energy function (J) with respect to the output weights is given by,

$$\frac{\partial^2 J(\mathbf{V})}{\partial {v_{lp}}^2} = \sum_{t=1}^{N} z_h^{p^t}\, \overline{z_h}^{k^t} = \sum_{t=1}^{N} |z_h^{p^t}|^2 > 0 \tag{4.39}$$

As the second derivative of the energy function $J(\mathbf{V})$ is positive, the following observations can be made from Eq. (4.39):

1. The function J is a convex function.
2. The output weight V^* obtained as a solution to the set of linear equations (Eq. (4.18)) is the weight corresponding to the minimum energy point of the energy function (J).

Using the *Theorem 1*, the solution for the system of equations in Eq. (4.18) can be determined as follows:

$$\overline{\mathbf{V}}^* = \mathbf{B}\mathbf{A}^{-1} \tag{4.40}$$

Applying the commutative law of multiplication of complex-valued conjugates,

$$\mathbf{V}^* = \overline{\mathbf{B}}\,\overline{\mathbf{A}}^{-1} \tag{4.41}$$

Thus the estimated output weights ($\mathbf{V}^*$) corresponds to the minimum energy point of the energy function.

The projection based learning algorithm of FCRN can be summarized as:
Given the training data set: $\left\{(\mathbf{z}^1, \mathbf{y}^1), ...(\mathbf{z}^t, \mathbf{y}^t), ..., (\mathbf{z}^N, \mathbf{y}^N)\right\}$
Step 1: Choose the number of hidden neurons: h and the random hidden layer parameters: $\mathbf{c}_k, \boldsymbol{\sigma}_k$; $k = 1, \cdots, h$
Step 1: Compute the hidden layer responses h_k^t using

$$z_h^{k^t} = sech\left(\boldsymbol{\sigma}_k^T \left(\mathbf{z}^t - \mathbf{c}_k\right)\right); \; k = 1, \cdots, h \tag{4.42}$$

Step 2: Compute the projection matrix $\mathbf{A}$ using

$$\mathbf{A}_{pk} = \sum_{t=1}^{N} z_h^{p^t} \overline{z_h}^{k^t}; \; p, k = 1, \cdots, h \tag{4.43}$$

Step 2: Compute the output matrix B using

$$\mathbf{B}_{lp} = \sum_{t=1}^{N} ln\left(\bar{y}_l^t\right) z_h^{p^t}; \; l = 1, \cdots, n; \; p = 1, \cdots, h \tag{4.44}$$

Step 3: Compute the optimum output weights using

$$\mathbf{W}^* = \overline{\mathbf{B}}\,\overline{\mathbf{A}}^{-1} \tag{4.45}$$

4.2 Summary

In this chapter, we have presented a projection based learning algorithm for a fully complex-valued relaxation network. For a given set of hidden layer neuron and their associated parameters, the projection based learning algorithm determines the output weights corresponding to the minimum energy point of an energy function which nonlinear, logarithmic and uses an explicit representation of both the magnitude and phase of the target and the predicted outputs. Using the Wirtinger calculus, the output weights of FCRN are determined as a solution to a nonlinear programming problem. The projection based learning algorithm computes the optimal output weights by converting the nonlinear programming problem to a problem of solving a set of linear equations. The output weights thus obtained are optimum and the training time required for learning is minimum.

In the next chapter, we evaluate the performances of the algorithms discussed in Chapters 2-4: FC-MLP, IC-MLP, FC-RBF, Mc-FCRBF and FCRN on a set of complex-valued function approximation problems.

References

1. Savitha, R., Suresh, S., Sundararajan, N., Saratchandran, P.: A new learning algorithm with logarithmic performance index for complex-valued neural networks. Neurocomputing 72(16-18), 3771–3781 (2009)
2. Wirtinger, W.: Zur formalen theorie der funktionen von mehr komplexen vernderlichen. Annals of Mathematics 97 (1927)
3. Yu, S.-S., Tsai, W.-H.: Relaxation by the Hopfield neural network. Pattern Recognition 25(2), 197–209 (1992)
4. Loss, D.V., de Castro, M.C.F., Franco, P.R.G., de Castro, E.C.C.: Phase transmittance RBF neural networks. Electronics Letters 43(16), 882–884 (2007)
5. Remmert, R.: Theory of Complex Functions. Springer, New York (1991)