The Complex Neural Networks

Melodi Çalışkan *

Master Student

Department of Computational Science and Engineering

Bogazici University

Bebek, Istanbul 34432 Turkey

January 14, 2016

*minemelodicaliskan@gmail.com

1 Introduction

In this study, complex-valued neural network structure and the back propagation algorithm will be discussed based on the usual neural networks as real valued.

The complex neural networks can be useful in various fields in which complex numbers are often used through Fourier Transformation such as image processing and pattern recognition.In addition, representing the phase and amplitude by complex numbers into neural networks suggests recoveries in the problems where both amplitude and phase is essential.

General neural networks and the back propagation algorithm in real domain will be briefly introduced and encountered problems in the derivation of the complex-valued back propagation algorithm, including the topics of complex gradient derivatives, Wirtinger Calculus and consequences of the Liouville's Theorem will be covered with the proposed solutions for the error function and the activation function of a neural network in complex domain.

2 Neural Networks

Artificial Neural Networks are an inspired approach from biological neurons for the data approximation and data classifications.

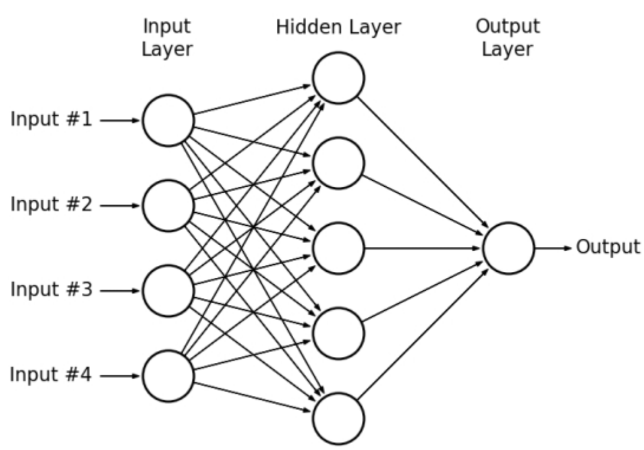

Figure 1: A single hidden layer network with n input nodes, m hidden nodes and 1 output node

In the feedforward neural networks the information moves only one direction as the sum of the multiplication of the nodes and the weights connecting the next layer calculated, it passes through the transfer (activation) function to determine the output of the layer.

The Back Propagation Algorithm

This algorithm looks for the minimum of the error function in weight space using the method of gradient descent.

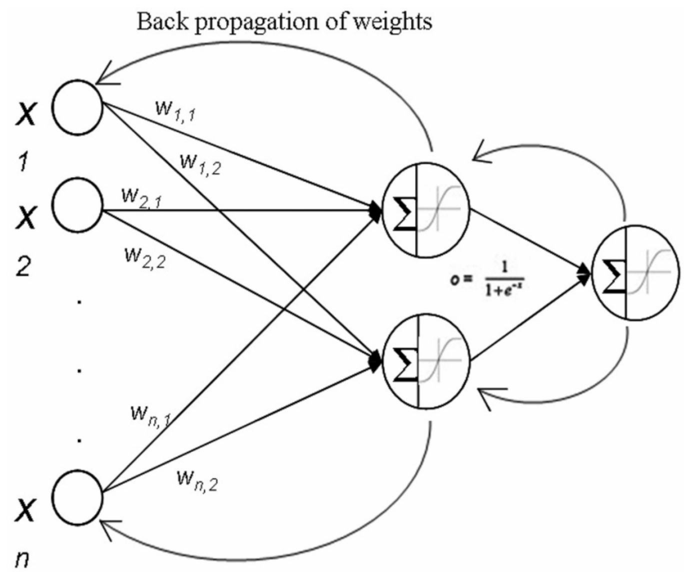

Figure 2: For given a training data set of input vector x and target output vector t , the algorithm back propagates the error by weighting it by the weights in the previous layer and the gradients of the associated activation functions.

After the back propagation the parameters are updated by using the calculated gradients.

The *activation function* f is non-linear, differentiable and bounded. The most preferred functions are :

1) Sigmoid function :$f(x) = \dfrac{1}{1+e^{-x}}$

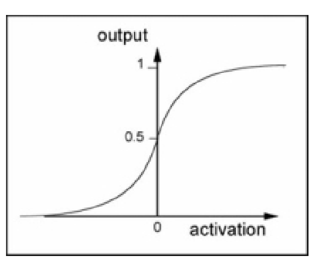

Figure 3

2) Hyperbolic tangent $\tanh(x) = (e^{x} - e^{-x})/(e^{x} + e^{-x})$

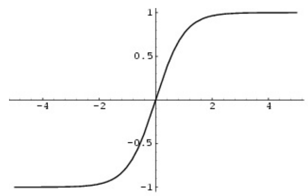

Figure 4

The Error Function

$$E = \frac{1}{2}\sum_{k\in K}(O_k - t_k)^2$$

where t_k is the target value of node k and O_k is the output value of node k which is obtained by weighted input value and activation function.

To minimize the error, the best combination of weights should be found and for this purpose it's rate of change with respect to given connective weights should be determined.
There exists two parts of the gradient computations depend on the layers:

1) Output layer node

2) Hidden layer node

Notation:
w_{jk} :The weight from hidden layer node j to the output layer node k
w_{ij} : The weight from input layer node i to the hidden layer node j
O_k: The output value of node k
f: The activation function
t_k: The target value of node k

In the following calculations *sigmoid activation function* will be used which has the derivative as:
$f'(x) = (\frac{1}{1+e^{-x}})(1-\frac{1}{1+e^{-x}}) = f(x)(1-f(x))$

Output Layer Node

$$\begin{aligned}
\frac{\partial E}{\partial \omega_{jk}} =& \frac{\partial}{\partial \omega_{jk}} \frac{1}{2} \sum_{k \in K} (O_k - t_k)^2 \\
&= (O_k - t_k) \frac{\partial}{\partial \omega_{jk}} O_k \qquad\qquad (*) \\
&= (O_k - t_k) \frac{\partial}{\partial \omega_{jk}} f(x_k) \\
&= (O_k - t_k) f^{'}(x_k) \frac{\partial}{\partial \omega_{jk}} x_k \\
&= (O_k - t_k) f^{'}(x_k) O_j \\
&= (O_k - t_k) O_k (1 - O_k) O_j \\
&\Longrightarrow \frac{\partial E}{\partial \omega_{jk}} = O_j \delta_k \quad where \quad \delta_k = (O_k - t_k) O_k (1 - O_k))
\end{aligned}$$

(*): In the second step the summation disappears because ω_{jk} is a specific weight connecting $j-th$ node to $k-th$ node from hidden and output layer respectively.

Hidden Layer Node

$$\begin{aligned}
\frac{\partial E}{\partial \omega_{ij}} =& \frac{\partial}{\partial \omega_{ij}} \frac{1}{2} \sum_{k \in K} (O_k - t_k)^2 \\
&= \sum_{k \in K} (O_k - t_k) \frac{\partial}{\partial \omega_{ij}} O_k \\
&= \sum_{k \in K} (O_k - t_k) \frac{\partial}{\partial \omega_{ij}} f(x_k) \\
&= \sum_{k \in K} (O_k - t_k) f^{'}(x_k) \frac{\partial x_k}{\partial \omega_{ij}} \\
&= \sum_{k \in K} (O_k - t_k) O_k (1 - O_k) \frac{\partial x_k}{\partial O_j} \frac{\partial O_j}{\partial \omega_{ij}} \\
&= \sum_{k \in K} (O_k - t_k) O_k (1 - O_k) \omega_{jk} \frac{\partial O_j}{\partial \omega_{ij}} \\
&= \frac{\partial O_j}{\partial \omega_{ij}} \sum_{k \in K} (O_k - t_k) O_k (1 - O_k) \omega_{jk} \\
&= f^{'}(x_j) \frac{\partial x_j}{\partial \omega_{ij}} \sum_{k \in K} (O_k - t_k) O_k (1 - O_k) \omega_{jk} \\
&= O_j (1 - O_j) O_i \sum_{k \in K} (O_k - t_k) O_k (1 - O_k) \omega_{jk} \\
&\Longrightarrow \frac{\partial E}{\partial \omega_{ij}} = O_i \delta_j \quad where \quad \delta_j = O_j (1 - O_j) \sum_{k \in K} \delta_k \omega_{jk}
\end{aligned}$$

Combining the results, back propagation algorithm updates the weights iteratively by: $\omega^{p+1} = \omega^p - b \nabla_\omega E(\omega^p)$

3 Complex Valued Neural Neural Network

The networks structure extends to complex domain straight forward as the inputs, weights, outputs are complex numbers in the complex-valued neural networks.

The Back Propagation Algorithm

The straight forward calculations as in the real-valued case the first problem of the computations occurs from the fact that the error function is no longer analytic since it becomes a mapping from complex space to real space.

The Error Function

Error function is real-valued in complex-valued back propagation also, in order to be able to compare the results as "bigger" or "smaller" since in complex domain such relations are not valid.

The complex error function can be defined as:

$$E = \sum_{k\in K} |O_k - t_k|^2 \sum_{k\in K} (O_k - t_k)\overline{(O_k - t_k)}$$

Since for any complex variable z;

$$|z|^2 = z\bar{z}$$

By setting $O_k - t_k = z_k$

It can be simplified as:

$$E(z,\bar{z}) = \sum_{k\in K} z_k \overline{z_k}$$

The problem appears in the definition of complex gradient, since the derivative for the complex conjugate $\bar{z}$ with respect to z is not defined since:

$$\frac{\partial \bar{z}}{\partial z} = \lim_{h\to 0} \frac{\overline{(z+h)} - \bar{z}}{h} = \begin{cases} 1 & h \in \mathbb{R} \\ -1 & h \in \mathbb{C} \end{cases}$$

To obtain complex gradient computations the error function can be viewed as a function of complex variable vector and its conjugate to use Wirtinger Calculus which enable us to take derivatives independently with respect to z and $\bar{z}$.

Wirtinger Derivatives

Consider a function $f : \mathbb{C} -> \mathbb{C}$:

$f(z) = u(z_r, z_{im}) + iv(z_r, z_{im})$ where $z = z_r + iz_{im}$

By rewriting $f(z)$ as $f(z_r, z_{im})$, it is well-known that then it can be considered as a function from $\mathbb{R}^2$ to $\mathbb{C}$ where z_rnd z_{im} can be written as: $z_r = (z+\bar{z})/2$and $z_{im} = (z-\bar{z})/2$

The *Wirtinger Derivatives* are defined as:

$$\frac{\partial f(z)}{\partial z} = 1/2(\frac{\partial f(z)}{\partial z_r} - i\frac{\partial f(z)}{\partial z_{im}})$$

$$\frac{\partial f(z)}{\partial \bar{z}} = 1/2(\frac{\partial f(z)}{\partial z_r} + i\frac{\partial f(z)}{\partial z_{im}})$$

The derivatives will be set to zero with respect to z or $\bar{z}$ to find the stationary points as:

1)If f is *complex differentiable* at α , then:

$$\frac{\partial f(\alpha)}{\partial z} = f'(\alpha)$$

$$\frac{\partial f(\alpha)}{\partial \bar{z}} = 0$$

2)If f is *conjugate-complex differentiable* at α , then:

$$\frac{\partial f(\alpha)}{\partial \bar{z}} = \overline{f(\alpha)}'$$

$$\frac{\partial f(\alpha)}{\partial z} = 0$$

Chain Rule For Complex Derivatives:

$$\frac{\partial f(g)}{\partial z} = \frac{\partial f}{\partial g}\frac{\partial g}{\partial z} + \frac{\partial f}{\partial \bar{g}}\frac{\partial \bar{g}}{\partial z}$$

$$\frac{\partial f(g)}{\partial \bar{z}} = \frac{\partial f}{\partial g}\frac{\partial g}{\partial \bar{z}} + \frac{\partial f}{\partial \bar{g}}\frac{\partial \bar{g}}{\partial \bar{z}}$$

<u>Note</u>:If $f(z) \in \mathbb{R}$ then;

$$\overline{(\frac{\partial f(\alpha)}{\partial z})} = \frac{\partial f(\alpha)}{\partial \bar{z}}$$

since $\overline{f(z)} = f(z)$

Using the formulations and the theorems above the real valued error function of vector valued complex variable z will be considered where the dependencies on the variable z and its conjugate $\bar{z}$ change by it's Wirtinger Derivatives. Then we will directly compute the derivatives with respect to complex argument, rather than calculating the individual real valued gradients with $\nabla_{\bar{z}} f(z)$

The Complex Activation Function

Before we begin the computations the complex activation function should be considered also, since in the complex domain the necessary properties; the boundedness and everywhere differentiability are only satisfied by constant functions which can be concluded from Liouville's Theorem. Then we have to choose either a bounded or an analytic function, i.e.; if the activation function is analytic then it should be unbounded and if it is bounded then it should be non-analytic.

There are two main proposed concepts to overcome this conflict:

1) Split-complex Activation Function: In this concept the function is derived from the convenient real-valued activation function as the real and imaginary part of it, therefore it is bounded everywhere in complex domain. Such activation functions make the data modeling easier with symmetry between two axes. The disadvantages of the split complex activation functions are that they can not conserve the complex data fully and they are not analytic, even though they are constructed from differentiable functions in the real domain. Computational problems in the differentiations can be again solved by Wirtinger Derivatives.

Example:

$$f(z) = tanh(Rez) + itanh(Imz)$$

2) Fully Complex Activation Function:

In this concept the function is derived by analytic continuation from real space to complex space. This provides the analyticity that is differentiability in a open set, therefore simplicity in the computations and ability to use the complex data fully.The disadvantage is that the real-bounded function becomes unbounded in the complex domain by the singularities and explodes the computations.This problem can be avoided by limiting the weight space.

Example:

$$f(z) = tanhz = (e^z - e^{-z})/(e^z + e^{-z})$$

has isolated singularities at every $((2n+1)/2)i\pi$, where n is an integer

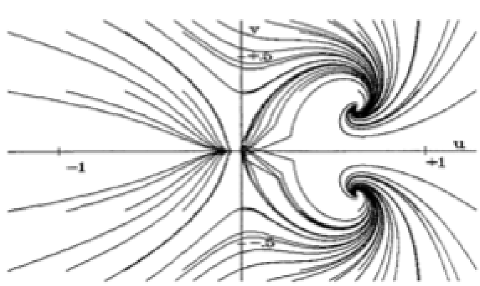

Figure 5: Activation trajectories for a single neuron with feedback value 1 and tanhz

If we modify the desired functional to avoid the singularities in the domain it becomes a convenient tool for gradient descent method.

In this paper *fully complex sigmoid activation function* is selected for the following calculations.

Output Layer Node

Conjugate Wirtinger Derivative:

$$\begin{aligned}
\frac{\partial E}{\partial \overline{\omega_{jk}}} &= \frac{\partial}{\partial \overline{\omega_{jk}}} \sum_{k\in K} (O_k - t_k)\overline{(O_k - t_k)} \\
&= \frac{\partial \overline{O_k}}{\partial \overline{\omega}_{jk}}(O_k - t_k) + \frac{\partial O_k}{\partial \overline{\omega_{jk}}}\overline{(O_k - t_k)} \\
&= \frac{\partial \overline{f(I_k)}}{\partial \overline{\omega}_{jk}}.(O_k - t_k) + 0.\overline{(O_k - t_k)} \\
&= (O_k - t_k) f^{'}(\overline{I_k}) \frac{\partial \overline{I_k}}{\partial \overline{\omega}_{jk}} \qquad\qquad (*) \\
&= (O_k - t_k)\overline{O_k}(1 - \overline{O_k})\overline{O_j} \\
&\Longrightarrow \frac{\partial E}{\partial \overline{\omega_{jk}}} = \overline{O_j}\delta_k \quad where \quad \delta_k = (O_k - t_k)\overline{O_k}(1 - \overline{O_k})
\end{aligned}$$

(*): $\overline{f(I_k)} = f(\overline{I_k})$ since

$$\overline{f(z)} = \overline{\left(\frac{e^z}{1+e^z}\right)} = \frac{e^{\overline{z}}}{1+e^{\overline{z}}} = f(\overline{z})$$

Hidden Layer Node

Conjugate Wirtinger Derivative:

$$
\begin{aligned}
\frac{\partial E}{\partial \overline{\omega_{ij}}} &= \frac{\partial}{\partial \overline{\omega_{ij}}} \sum_{k \in K} (O_k - t_k)\overline{(O_k - t_k)} \\
&= \frac{\partial O_k}{\partial \overline{\omega_{ij}}} \overline{(O_k - t_k)} + \frac{\partial \overline{O_k}}{\partial \overline{\omega}_{ij}} (O_k - t_k) \\
&= \sum_{k \in K} 0.\overline{(O_k - t_k)} + \frac{f(\overline{I_k})}{\partial \overline{\omega}_{ij}} (O_k - t_k) \\
&= \sum_{k \in K} \frac{\partial f(\overline{I_k})}{\partial \overline{\omega}_{ij}} (O_k - t_k) \\
&= \sum_{k \in K} (O_k - t_k) f^{'}(\overline{I_k}) \frac{\partial \overline{I_k}}{\partial \overline{\omega}_{ij}} \\
&= \sum_{k \in K} (O_k - t_k) \overline{O_k} (1 - \overline{O_k}) \frac{\partial \overline{I_k}}{\partial \overline{O}_j} \frac{\partial \overline{O}_j}{\partial \overline{\omega}_{ij}} \\
&= \frac{\partial \overline{O}_j}{\partial \overline{\omega}_{ij}} \sum_{k \in K} (O_k - t_k) \overline{O_k} (1 - \overline{O_k}) \overline{\omega}_{jk} \\
&= f^{'}(\overline{I_j}) \frac{\partial \overline{I_j}}{\partial \overline{\omega}_{ij}} \sum_{k \in K} (O_k - t_k) \overline{O_k} (1 - \overline{O_k}) \overline{\omega}_{jk} \\
&= \overline{O_j} (1 - \overline{O_j}) \frac{\partial \overline{I_j}}{\partial \overline{\omega}_{ij}} \sum_{k \in K} (O_k - t_k) \overline{O_k} (1 - \overline{O_k}) \overline{\omega}_{jk} \\
&= \overline{O_j} (1 - \overline{O_j}) \overline{O_i} \sum_{k \in K} (O_k - t_k) \overline{O_k} (1 - \overline{O_k}) \overline{\omega}_{jk} \\
\Longrightarrow \frac{\partial E}{\partial \overline{\omega_{ij}}} &= \overline{O_i} \delta_j \quad where \quad \delta_j = \overline{O_j} (1 - \overline{O_j}) \sum_{k \in K} \delta_k \overline{\omega_{jk}}
\end{aligned}
$$

Combining the results complex back propagation algorithm updates the weights iteratively by:

$\omega^{p+1} = \omega^p - b \nabla_{\overline{\omega}} E(\omega^p)$

4 Conclusion

The neural network structure and the back propagation algorithm investigated in complex domain by extending from the real-valued neural networks. Wirtinger derivatives and associated theorems and the convenient concepts of the complex activation functions covered. Ongoing studies about the complex neural networks exist and the author will continue to research about the possible complex activation functions in the neural networks.

References

[1] Thomas L. Clarke (1990) *Generalization of Neural Networks to Complex Plane* IJCNN International Joint Conference on San Diego, CA, USA
10.1109/IJCNN.1990.137751

[2] Xiaodong Liu, Dongpo Xu, and Ying Zhang (2014) *Convergence analysis of fully complex backpropagation algorithm based on Wirtinger calculus*. Cogn Neurodyn
10.1007/s11571-013-9276-7

[3] Tülay Adalı, Taehwan Kim (2003) *Approximation by Fully Complex Multilayer Perceptrons*
10.1162/089976603321891846