CLASSIFICATION OF POLARIMETRIC SAR DATA BY COMPLEX VALUED NEURAL NETWORKS

Ronny Hänsch and Olaf Hellwich

Berlin Institute of Technology
Computer Vision and Remote Sensing
Berlin, Germany
rhaensch@fpk.tu-berlin.de; hellwich@cs.tu-berlin.de

KEY WORDS: PolSAR, classification, machine learning, complex valued neural networks

ABSTRACT:

In the last decades it often has been shown that Multilayer Perceptrons (MLPs) are powerful function approximators. They were successfully applied to a lot of different classification problems. However, originally they only deal with real valued numbers. Since PolSAR data is a complex valued signal this paper propose the usage of Complex Valued Neural Networks (CVNNs), which are an extension of MLPs to the complex domain. The paper provides a generalized derivation of the complex backpropagation algorithm, mentions regarding problems and possible solutions and evaluate the performance of CVNNs by a classification task of different landuses in PolSAR images.

1 INTRODUCTION

Synthetic Aperture Radar (SAR) achieved more and more importance in remote sensing within the last years. Nowadays a great amount of SAR data is provided by a lot of air- and spaceborne sensors. SAR represents an important source of remote sensing data, since the image acquisition is independent from daylight and less influenced by weather conditions. Polarimetric SAR (PolSAR) possesses all those advantages, but also add polarimetry as a further very meaningful information. Several scattering mechanisms can be better or even only distinguished by usage of polarimetry. Therefore, most contemporary sensors are able to provide PolSAR data.

The increasing amount of remote sensing images available nowadays requires automatic methods, which robustly analyse those data. Especially segmentation and classication of PolSAR images are of high interest as a first step to a more general image interpretation. However, this aim is very challenging due to the difficult characteristics of SAR imagery.

In supervised learning schemes the user provides a training set, consisting of the wanted system output for several given system inputs. There are a lot of different supervised learning methods, such as support vector machines, linear regression, radial basis function networks and multilayer perceptrons (MLPs). Some of those have been already applied to classication of SAR or PolSAR images (Fukuda and Hirosawa, 2001). However, all of them are designed for real valued data, which makes preprocessing steps necessary, when dealing with complex valued PolSAR data. This leads to a loss of available information. Furthermore, a lot of classification schemes for PolSAR data make parametric assumptions about the underlying distributions, which tend to fail with contemporary high-resolution SAR data. Therefore, this paper evaluates the usage of complex valued neural networks (CVNNs), which are an extension of well studied MLPs to the complex domain. MLPs are known to be powerful function approximators with good convergence and classification performance. CVNNs have already shown their applicability to SAR data (Hirose, 2006), but by the best of the authors knowledge have been never used in classication of PolSAR data.

In contrast to MLPs, which are real functions of real arguments with real coefficients, CVNNs deal with complex valued inputs by using complex weights. This extension seems to be straightforward, but is complicated by restrictions of derivatives in the complex domain.

The next chapter describes the dynamics of a CVNN. Problems and solutions regarding the extension to the complex domain are discussed and the used variation of the backpropagation algorithm is explained. In Chapter 3 the performance of CVNNs is evaluated by a classification task.

2 COMPLEX VALUED NEURAL NETWORKS

2.1 Dynamics

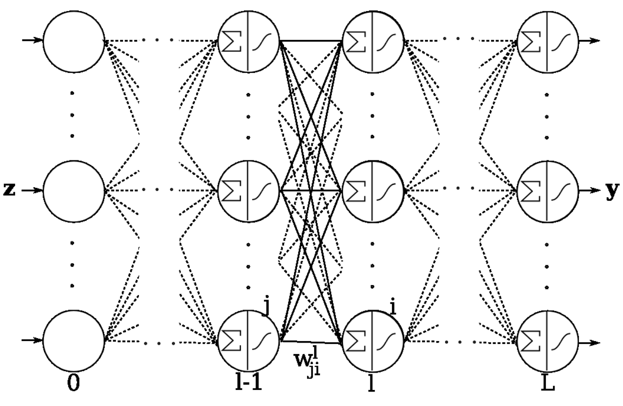

Figure 1: CVNN

Similar to MLPs a CVNN consists of several simple units, called neurons, which are ordered in a certain number L of different layers (see Fig.1). The first layer is called input layer ($l = 0$), the last one output layer ($l = L$) and all layers in between hidden layers. Only neurons in subsequent layers are connected and a weight w is associated with each of those connections. The activation y_i^l of the i-th neuron in layer l in terms of the net input h_i^l of this neuron is calculated as:

$$y_i^l = \begin{cases} z_i & \text{if } l = 0, \\ f_i^l(h_i^l) & \text{else.} \end{cases} \tag{1}$$

$$h_i^l = \sum_{j=1}^{N_{l-1}} y_j^{l-1} \cdot {w_{ji}^l}^* \tag{2}$$

where h_i^l, y_i^l, z_i, $w_{ji}^l \in \mathbb{C}$, N_{l-1} is the number of neurons in the previous layer and $(\cdot)^*$ means complex conjugation. The function $f_i^l(\cdot)$ is called activation function and is mostly chosen to perform a nonlinear transformation of the weighted input. The sigmoid functions given by Eq.3,4 are the most common choice for MLPs, since they are differentialable in every point (analytic) and bounded.

$$f(x) = \tanh(x) \tag{3}$$

$$f(x) = \frac{1}{1+e^{-x}} \tag{4}$$

Possible activation functions for CVNNs are discussed in section 2.3

As CVNNs are supervised learning schemes, a user provided data set $\mathbf{D}$ exists:

$$\mathbf{D} = \left\{ (\mathbf{z}, \tilde{\mathbf{y}})^{(\alpha)} \right\}_{\alpha=\mathbf{1},..,\mathbf{P}} \tag{5}$$

$$\mathbf{z} \in \mathbb{C}^{N_0}, \tilde{\mathbf{y}} \in \mathbb{C}^{N_L} \tag{6}$$

The goal is to find a specific set of weights $\{w_{ji}^l\}$ for which the system output $\mathbf{y}^L$ for a given input $\mathbf{z}$ equals the target value $\tilde{\mathbf{y}}^{(\alpha)}$. Therefore the minimum of an error function $E_{\{w\}}(\mathbf{D})$ is searched:

$$E_{\{w\}}(\mathbf{D}) = \frac{1}{P} \sum_{\alpha=1}^{P} err\left(\mathbf{y}^L(\mathbf{z}^{(\alpha)}), \tilde{\mathbf{y}}^{(\alpha)} \right) \tag{7}$$

where $err(\cdot)$ is the individual error of each data point.

2.2 Complex Valued Backpropagation Algorithm

The mostly used method for adjusting the free parameters of a MLP, named backpropagation, relies on the gradient of the error function. The same idea is utilized for CVNNs. The following derivative of the complex backpropagation algorithm is geared to the derivation given in (Yang and Bose, 2005), but does not require a specific activation or error function. Therefore, it results in a more general learning rule, which can easily be adapted to different choices of those functions.

The derivatives of complex functions are obtained according to the complex chain rule

$$\frac{\partial g(h(\mathbf{z}))}{\partial \mathbf{z}} = \frac{\partial g(h(\mathbf{z}))}{\partial h(\mathbf{z})} \cdot \frac{\partial h(\mathbf{z})}{\partial \mathbf{z}} + \frac{\partial g(h(\mathbf{z}))}{\partial h(\mathbf{z})^*} \cdot \frac{\partial h(\mathbf{z}))^*}{\partial \mathbf{z}} \tag{8}$$

the Generalized Complex Derivative

$$\frac{\partial g(\mathbf{z})}{\partial \mathbf{z}} = \frac{1}{2}\left(\frac{\partial g(\mathbf{z})}{\partial \Re \mathbf{z}} - j \frac{\partial g(\mathbf{z})}{\partial \Im \mathbf{z}} \right) \tag{9}$$

and the Conjugate Complex Derivative (equals zero if $g(\mathbf{z})$ is analytic)

$$\frac{\partial g(\mathbf{z})}{\partial \mathbf{z}^*} = \frac{1}{2}\left(\frac{\partial g(\mathbf{z})}{\partial \Re \mathbf{z}} + j \frac{\partial g(\mathbf{z})}{\partial \Im \mathbf{z}} \right) \tag{10}$$

where $j = \sqrt{-1}$ is the imaginary unit.

For more detailed information about analysis of complex function see (Haykin, 2002).

Given a specific data set as in Eq.5 the minimum of the error function is searched by iteratively adjusting the weights according to:

$$w_{ji}^l(t+1) = w_{ji}^l(t) + \tilde{\mu} \Delta w_{ji}^l(t) \tag{11}$$

where $\tilde{\mu}$ is the learning rate and the current change of the weights $\Delta w_{ji}^l(t)$ depends on the negative gradient of the error function. As stated by (Haykin, 2002) this can be calculated as:

$$\Delta w_{ji}^l(t) = -\nabla E_{\{w\}}(\mathbf{D}) = -2 \cdot \frac{\partial err_{\{w\}}(\mathbf{D})}{\partial {w_{ji}^l}^*} \tag{12}$$

$$\frac{\partial err_{\{w\}}(\mathbf{D})}{\partial {w_{ji}^l}^*} = \frac{1}{P} \sum_{\alpha=1}^{P} \frac{\partial err\left(\mathbf{y}^L(\mathbf{z}^{(\alpha)}), \tilde{\mathbf{y}}^{(\alpha)} \right)}{\partial {w_{ji}^l}^*} \tag{13}$$

$$\frac{\partial err}{\partial {w_{ji}^l}^*} = \underbrace{\frac{\partial err}{\partial h_i^l}}_{\delta_i^l} \cdot \frac{\partial h_i^l}{\partial {w_{ji}^l}^*} + \underbrace{\frac{\partial err}{\partial {h_i^l}^*}}_{{\delta_i^l}^*} \cdot \frac{\partial {h_i^l}^*}{\partial {w_{ji}^l}^*} \tag{14}$$

$$\frac{\partial h_i^l}{\partial {w_{ji}^l}^*} = y_j^{l-1} \tag{15}$$

$$\frac{\partial {h_i^l}^*}{\partial {w_{ji}^l}^*} = 0 \tag{16}$$

$$\delta_i^l = \frac{\partial err}{\partial h_i^l} = \frac{\partial err}{\partial y_i^l} \cdot \frac{\partial y_i^l}{\partial h_i^l} + \frac{\partial err}{\partial y_i^{l*}} \cdot \frac{\partial {y_i^l}^*}{\partial h_i^l} \tag{17}$$

$$\delta_i^{l*} = \frac{\partial err}{\partial h_i^{l*}} = \frac{\partial err}{\partial y_i^l} \cdot \frac{\partial y_i^l}{\partial h_i^{l*}} + \frac{\partial err}{\partial y_i^{l*}} \cdot \frac{\partial {y_i^l}^*}{\partial h_i^{l*}} \tag{18}$$

If the current layer l is the output layer ($l = L$) only δ_i^L is needed and can be directly calculated if the complex derivatives of the activation function of the output layer $\left(\frac{\partial y_i^l}{\partial h_i^l}, \frac{\partial {y_i^l}^*}{\partial h_i^l} \right)$ and the derivatives of the individual error function $\left(\frac{\partial err}{\partial y_i^l}, \frac{\partial err}{\partial {y_i^l}^*} \right)$ are known.

If the current layer l is one of the hidden layers ($0 < l < L$) the error is backpropagated by:

$$\frac{\partial err}{\partial y_i^l} = \sum_{r=1}^{N_{l+1}} \left(\frac{\partial err}{\partial h_r^{l+1}} \cdot \frac{\partial h_r^{l+1}}{\partial y_i^l} + \frac{\partial err}{\partial h_r^{l+1*}} \cdot \frac{\partial {h_r^{l+1}}^*}{\partial y_i^l} \right) \tag{19}$$

$$\frac{\partial err}{\partial y_i^{l*}} = \sum_{r=1}^{N_{l+1}} \left(\frac{\partial err}{\partial h_r^{l+1}} \cdot \frac{\partial h_r^{l+1}}{\partial y_i^{l*}} + \frac{\partial err}{\partial h_r^{l+1*}} \cdot \frac{\partial {h_r^{l+1}}^*}{\partial y_i^{l*}} \right) \tag{20}$$

$$\tag{21}$$

According to Eq.2 one obtains:

$$\frac{\partial h_r^{l+1}}{\partial y_i^l} = w_{ir}^{l+1*} \tag{22}$$

$$\frac{\partial h_r^{l+1*}}{\partial y_i^l} = 0 \tag{23}$$

$$\frac{\partial h_r^{l+1}}{\partial y_i^{l*}} = 0 \tag{24}$$

$$\frac{\partial h_r^{l+1*}}{\partial y_i^{l*}} = w_{ir}^{l+1} \tag{25}$$

Using Eq.19-25 Eq.17 and 18 become to:

$$\delta_i^l = \sum_{r=1}^{N_{l+1}} \left(\delta_r^{l+1} \cdot w_{ir}^{l+1*} \cdot \frac{\partial y_i^l}{\partial h_i^l} + \delta_r^{l+1*} \cdot w_{ir}^{l+1} \cdot \frac{\partial y_i^{l*}}{\partial h_i^l} \right) \quad (26)$$

$$\delta_i^{l*} = \sum_{r=1}^{N_{l+1}} \left(\delta_r^{l+1} \cdot w_{ir}^{l+1*} \cdot \frac{\partial y_i^l}{\partial h_i^{l*}} + \delta_r^{l+1*} \cdot w_{ir}^{l+1} \cdot \frac{\partial y_i^{l*}}{\partial h_i^{l*}} \right) \quad (27)$$

Using Eq.13-18 and Eq.26,27 the learning rule in Eq.11 becomes to:

$$w_{ji}^l(t+1) = w_{ji}^l(t) - \mu \sum_{\alpha=1}^{P} \delta_i^l \cdot y_j^{l-1} \quad (28)$$

$$\text{where } \mu = \frac{2 \cdot \tilde{\mu}}{P} \text{ and} \quad (29)$$

$$\delta_i^l = \begin{cases} \frac{\partial err}{\partial y_i^l} \cdot \frac{\partial y_i^l}{\partial h_i^l} + \frac{\partial err}{\partial y_i^{l*}} \cdot \frac{\partial y_i^{l*}}{\partial h_i^l} & , l = L \\ \sum_{r=1}^{N_{l+1}} \left(\delta_r^{l+1} w_{ir}^{l+1*} \frac{\partial y_i^l}{\partial h_i^l} + \delta_r^{l+1*} w_{ir}^{l+1} \frac{\partial y_i^{l*}}{\partial h_i^l} \right) & , l < L \end{cases} \quad (30)$$

$$\delta_i^{l*} = \begin{cases} \frac{\partial err}{\partial y_i^l} \cdot \frac{\partial y_i^l}{\partial h_i^{l*}} + \frac{\partial err}{\partial y_i^{l*}} \cdot \frac{\partial y_i^{l*}}{\partial h_i^{l*}} & , l = L \\ \sum_{r=1}^{N_{l+1}} \left(\delta_r^{l+1} w_{ir}^{l+1*} \frac{\partial y_i^l}{\partial h_i^{l*}} + \delta_r^{l+1*} w_{ir}^{l+1} \frac{\partial y_i^{l*}}{\partial h_i^{l*}} \right) & , l < L \end{cases} \quad (31)$$

2.3 Activation Function

To apply the learning rule given by Eq.28 the derivatives of the activation functions are needed. As stated above the activation functions should be bounded and differentiable in every point. However, in the complex domain only constant functions fulfill both criteria as stated by Liouville's theorem. Due to this fact CVNNs were not considered as robust and well performing learning methods until few years ago.

One of the first ideas to deal with complex information was to use two different MLPs, which were separably trained on real and imaginary parts or amplitude and phase. Obviously this approach cannot use the full complex information available in PolSAR data.

Recently more sophisticated ideas have been published, which make use of either bounded but not analytic or analytic but not bounded activation functions (Yang and Bose, 2005, Kim and Adali, 2003). The complex $\tanh$-function for example (see Fig.2) is analytic, but not bounded since it has singularities at every $(\frac{1}{2}+n)\pi i, n \in \mathbb{N}$. However, if the input was properly scaled, the weights were initilized with small enough values and one takes care that they do not exceed a certain bound during training, the net-input of an neuron will not approach this critical areas.

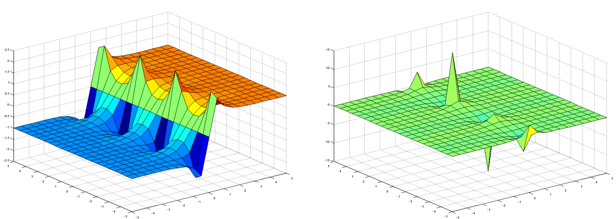

Figure 2: real (left) and imaginary (right) part of $\tanh(z)$

On the other side if the saturation properties of real valued activation functions are considered, a function like shown in Fig.3 seems to be a more reasonable extension of real valued sigmoidal functions in the complex domain, since it shows a similar saturation.

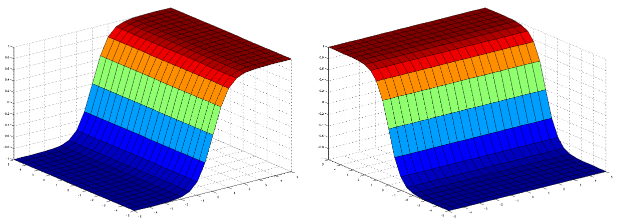

Figure 3: real (left) and imaginary (right) part of split-$\tanh(z)$

This function is called split-$\tanh$ and is defined by:

$$f(z) = \tanh(\Re z) + i \tanh(\Im z) \quad (32)$$

It is bounded everywhere in $\mathbb{C}$. Although it is not analytic the Generalized Complex Derivative (Eq.9) and Conjugate Complex Derivative (Eq.10) exist. Therefore it was used in this paper.

2.4 Error Functions

The other derivatives needed by Eq.28 are the complex derivatives of the error function. In the most cases the quadratic error defined by Eq.33 is used when learning with backpropagation. However, it was reported in (Prashanth et.all, 2002) that other error functions can significantly improve the classification result. The error functions $err\left(\mathbf{y}^L(\mathbf{z}^{(\alpha)}), \tilde{\mathbf{y}}^{(\alpha)}\right)$ tested within this paper are:

- Complex Quadratic Error Function $CQErr$:

$$\sum_{i=0}^{N_L} \frac{1}{2} \epsilon_i \epsilon_i^* \quad (33)$$

- Complex Fourth Power Error Function $CFPErr$:

$$\sum_{i=0}^{N_L} \frac{1}{2} \left(\epsilon_i \epsilon_i^*\right)^2 \quad (34)$$

- Complex Cauchy Error Function $CCErr$:

$$\sum_{i=0}^{N_L} \frac{c^2}{2} \ln\left(1 + \frac{\epsilon_i \epsilon_i^*}{c^2}\right) \quad (35)$$

  where c is a constant which was set to unity.

- Complex Log-Cosh Error Function $CLCErr$:

$$\sum_{i=0}^{N_L} \ln(\cosh(\epsilon_i \epsilon_i^*)) \quad (36)$$

where $\epsilon_i = \tilde{y}_i^{(\alpha)} - y_i^L\left(\mathbf{z}^{(\alpha)}\right)$.

3 RESULTS AND DISCUSSION

To determine the performance of CVNNs they were utilized to solve a classification task in this paper.

3.1 Used Data

PolSAR uses microwaves with different polarisations to measure the distance to ground and the reflectance of a target. The most common choice are two orthogonal linear polarisations, namely horizontal (H) and verticular (V) polarisation. Those settings result in a four dimensional vector, that can be reduced to a three dimensional vector $\mathbf{k}$ under the assumption of reciprocity of natural targets:

$$\mathbf{k} = (S_{HH}, S_{HV}, S_{VV}) \tag{37}$$

where S_{TR} is the complex valued measurement of the signal backscattered from the surface using the indicated polarisation during transmitting (T) and receiving (R).

There are several targets, which cannot be fully described by a single scattering vector. Therefore, the complex sample covariance matrix given by

$$\mathbf{C} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{k}_i \tag{38}$$

is often a better description of PolSAR data and was used as input to all CVNNs in the following analysis.

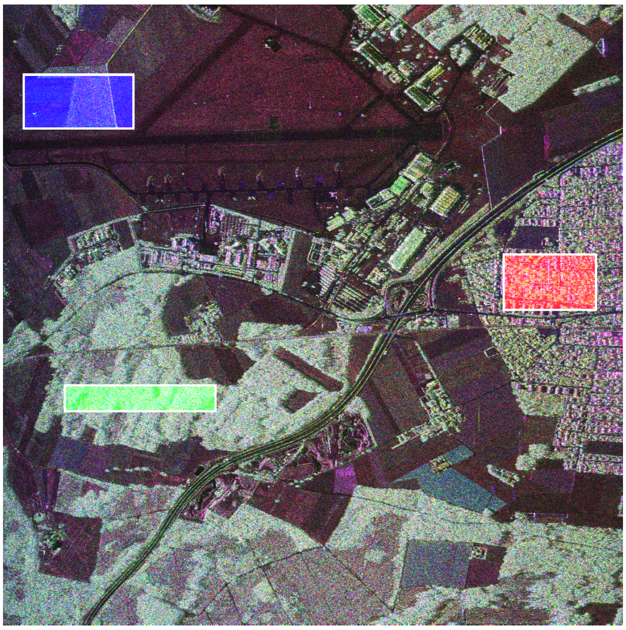

Figure 4: ESAR image over Oberpfaffenhofen with marked training areas

The image in Fig.4 is a color representation of the PolSAR data used by the following analysis. It was taken by E-SAR sensor over Oberpfaffenhofen at L-band and consists of regions of forests and fields as well as urban areas. The trained CVNNs have to distinguish between those three types of landcover. Training areas, which were labelled manually, are marked by colored rectangles within the image.

The target value $\tilde{\mathbf{y}}$ of class c is a three dimensional complex vector whose i-th component is defined by:

$$\tilde{y}_i = (-1)^d \cdot (1 + j) \tag{39}$$

$$d = \begin{cases} 0 \text{ if } i = c \\ 1 \text{ if } i \neq c \end{cases} \tag{40}$$

3.2 Performance regarding to different Error Functions

Table1 summarizes the achieved classification accuracy. Each column shows the average percentage of misclassification on previous unseen data using different error functions given by Eq.33-36. Different net architectures are orderd in different rows, beginning with the most simple case of only one layer at the top of Table1 to a three layer architecture (9 input neurons, 10 neurons for each of the hidden layers and 3 output neurons) at the bottom.

	CQErr	CFPErr	CCErr	CLCErr
9-3	13.29%	47.70%	14.57%	21.03%
9-5-3	12.40%	41.63%	15.68%	12.19%
9-10-3	11.23%	24.17%	15.68%	13.05%
9-10-10-3	9.41%	14.67%	9.87%	10.82%

Table 1: CVNN classification error for different error functions and net architectures

Although the results are very similar for more complex architectures, the CVNNs using the Complex Quadratic Error Function and the Complex Cauchy Error Function were able to achieve good classifications with only one layer.

3.3 Comparison with MLP

As MLPs cannot deal with the original complex signal different sets of features have to be chosen. Two different features were used to train MLPs and to compare their classification performance with those of CVNNs:

- amplitudes of the complex components of the sample covariance matrix $\mathbf{C}$:

$$f_1 = |\mathbf{C}| \tag{41}$$

- eigenvalues e_i and diagonal elements c_{ii} of the sample covariance matrix $\mathbf{C}$ (both are real valued, since $\mathbf{C}$ is hermitian):

$$f_2 = (e_1, e_2, e_3, c_{11}, c_{22}, c_{33}) \tag{42}$$

The quadratic error function was used together with a $\tanh$-activation function. The MLP consisted of one hidden layer with ten neurons.

	f_1	f_2
N_0-10-3	54.86%	57.13%

Table 2: MLP classification errors for different features

The classification results as average percentage of misclassification are summarized in Table2. They clearly show that an MLP trained with the features above is unable to solve the classification task. Note that they differ only slightly from a random label assignment to all data points.

4 CONCLUSIONS

The paper provides a generalized derivation of the backpropagation algorithm in the complex domain. The derived learning rule can easily adapted to different activation and error functions. The performance of CVNNs was evaluated. They appear to be superior to real valued MLPs regarding the specific classification task, since they achieved a good performance on average where MLPs nearly completely fail. Although the split-$\tanh$ activation function is non-analytic it is possible to obtain a meaningful gradient using Complex Generalized Derivative and Complex Conjugate Derivative. However, the behavior of CVNNs under the usage of different activation functions should be investigated, what will be part of the future work of the authors.

REFERENCES

Fukuda, S. and Hirosawa, H., 2001. Support vector machine classification of land cover: application to polarimetric SAR data. IEEE 2001 International Geoscience and Remote Sensing Symposium, 2001. IGARSS'01 vol.1, pp.187-189

Hirose, A., 2006 Land-surface classification with unevenness and reflectance taken into consideration. Studies in Computational Intelligence, Complex-Valued Neural Networks pp.89-97

Haykin, S., 2002 Adaptive Filter Theory Prentice Hall, Upper Saddle River,NJ, 4th edition, 2002 pp.798

Prashanth, A., Kalra, P.K. and Vyas, N.S., 2002 Surface classification using ANN and complex-valued neural network Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP '02 vol.3, pp.1094-1098

Yang, C-C. and Bose, N.K., 2005 Landmine detection and classification with complex-valued hybrid neural network using scattering parameters dataset IEEE transactions on neural networks vol.16, no.3, pp.743-753

Kim, T. and Adali, T., 2003 Approximation by fully complex multilayer perceptrons Neural computation 2003 vol.15, no.7, pp.1641-1666