ISSN 2238-1007

UNCERTAINTIES 2012

Proceedings of the 1st International Symposium on Uncertainty Quantification and Stochastic Modeling
February 26th to March 2nd, 2012, Maresias, São Sebastião, SP, Brazil

COMPARISONS OF TWO LEARNING STRATEGIES FOR A SUPERVISED NEURAL NETWORK

Fábio Dall Cortivo[1], **Ezzat S. Chalhoub**[2], **Haroldo F. de Campos Velho**[2]

[1] Instituto Nacional de Pesquisas Espaciais (INPE), Programa de Pós-Graduação em Computação Aplicada (CAP),
fabio.cortivo@lac.inpe.br
[2] Instituto Nacional de Pesquisas Espaciais (INPE), Laboratório Associado de Computação e Matemática Aplicada (LAC),
{ezzat,haroldo}@lac.inpe.br

Abstract. *Artificial neural networks can be used to solve inverse problems. One relevant problem in hydrologic optics is the estimatation of the single scattering albedo from the emitted surface radiation. The multi-layer perceptron (MLP) can be applied to determine the albedo from the measured radiation. The MLP is designed with one hidden layer, where the activation employs the sigmoid function, with backpropagation for set-upping the network parameters. Using the generalized delta rule for the learning process to determine the weight connections, the neural inverse operator (ANN-1) produces good results with 20 inputs (10 incident beams, and 10 emitted beams) and 40 neurons in the hidden layer in two different groups of neurons (30 and 10), with two different parameters for the sigmoid function. The second scheme for training the neural estimator applies the quasi-Newton optimization. For the last strategy, the final neural inverse operator (ANN-2) has 10 inputs (emitted radiation) and 20 neurons in the hidden layer in two different groups of neurons (15 and 5). The measured data were emulated considering five levels of noise. For the generalization test, the ANN-1 and ANN-2 operators obtained 100% of correct answers for the noiseless observational data. For noisy data, the ANN-1 operator obtained 94% of correct answers, while the ANN-2 operator obtained 100% of correct answers. The main difference between these two ANNs is the training method, and the number of neurons in the input and the hidden layer.*

Keywords. *Inverse Problems, Artificial Neural Networks, Backpropagation, Quasi-Newton Method, Hydrologic Optics*

1 INTRODUCTION

Many methodologies of inversion in radiative transfer have been developed in various areas in science and engineering (McCormick, 1992), and a revision of these methods and their applications were performed by McCormick (2001), where the author relates about inverse problems that can explicitly be solved (McCormick, 2004).

A detailed review in inverse Hydrologic Optics can be found in Gordon (2002), and in other works where: Stephany (1998) estimated the bioluminescence internal source using the invariant embedding method (Mobley, 1994); Stephany et al. (2000) estimated the absorption and scattering coefficients also using the invariant embedding method; Chalhoub et al. (2000) estimated the phase function using the S_N method (Chandrasekhar, 1950); Retamoso et al. (2002) estimated the boundary conditions using the LTS_N method (Vilhena & Barichello, 1991); Chalhoub & Campos Velho (2001, 2002, 2003) estimated the phase function, the single scattering albedo, the optical depth and the internal sources, using the AS_N method (Chalhoub, 1997); and finally Souto (2006) estimated the chlorophyll concentration using the LTS_N method.

In the above refereced works, an implicit approach was used to attack the inverse problems, in order to search for the minimal to a functional defined as the quadratic difference sum between the observed data (or synthetic data generated by a computer) and the calculated data using adopted inverse formulations. The search for this minimal was performed by a deterministic method, except for the last work, where the minimal was obtained by the Ant Colony Optimization stochastic technique (Dorigo, 1992).

This type of approach produced good results as noted in the conclusions of the cited references, although, depending on the solved problem, the computational time consumed to find the solution was relatively large due to the extensive and overly refined mathematical formalisms of the direct problem. Furthermore, to obtain the inverse solution, the direct problem must be solved many times, making it necessary to use parallel processing in order to reduce the computational time involved, as emphasized in Souto (2006).

Regarding the consumed computational time, specially for solving problems dealing with the solution of the Radiative Transfer Equation (RTE), a new strategy has lately become dominant: the Artificial Neural Network (ANN) strategy. Due to the fact that the behavior of an ANN is based on the human brain, an ANN strategy presents many advantages over other strategies, such as learning capacity, self-organization, fault tolerance, flexibility and, the most important of all, is its capacity to produce real-time responses. A detailed information of these qualities can be found in Fausett (1994) and Haykin (1999).

In fact, the works (Bokar, 1999; Soeiro et al., 2004; Soeiro & Silva Neto, 2006; Chalhoub et al., 2007; Oliveira, 2010; Cortivo et al., 2010) showed that the utilization of the ANN strategy in solving inverse radiative transfer problems is quite promising. In this case, the ANNs acted as inverse operators, emulating the mathematical formalisms involved in the direct/inverse problems. So, the optimization problem within the inverse problem was exchanged with the network training, which can also be considered as an optimization problem. However, in many cases, the network training can be

ISSN 2238-1007

a slow process, getting stuck in the local minimum, or even requiring many processes in order to obtain a response within an acceptable error interval.

There exist different network topologies and one of them is the Multi-Layer Perceptron (MLP), where the common training algorithm is the backpropagation. This algorithm, which can present the difficulties mentioned above, is a supervised type and is based on the gradient of the error function. A supervised type algorithm requires the knowledge of the input pattern that is associated with the network output. In Cortivo et al. (2010), the single scattering albedo was estimated, utilizing a MLP neural network with the backpropagation algorithm error and using the generalized delta rule (Haykin, 1999) for the correction of the synaptic weights. In that work, good results were obtained regarding the calculated correct answers, which were above 90% in all considered cases, however the network presented some difficulties in recognizing some patterns (radiances) that were generated with the albedo $\varpi \in [0.2, 0.5]$.

Knowing of the problems associated with the training of a neural network with the backpropagation algorithm, and the difficulty to correctly classify some patterns, we used in this work a strategy that utilizes the quasi-Newton method (Dennis & Moré, 1977) as a training algorithm. Another advantages of this method is the possibility of formulating the optimization problem (training) with restrictions, *i.e.*, the variables to be optimized are restricted to pre-established maximum and minimum values, in order to prevent the connection weights to reach undesired high values and thus avoid the neurons saturation. Moreover, the quasi-Newton method has a better convergence when compared to the backpropagation algorithm (Watrous, 1988; Setiono & Hui, 1995).

2 THE PROBLEM

The Radiative Transfer Equation (RTE) is a mathematical model which describes the interaction of the radiation field in the medium that absorbs, emits and scatters it. The RTE for problems in plane-parallel geometry, with azimuthal dependence and without internal sources is given by,

$$\mu \frac{\partial}{\partial \tau} I(\tau, \vec{\xi}) + I(\tau, \vec{\xi}) = \varpi \int_{-1}^{1} \int_{0}^{2\pi} \beta(\vec{\xi}; \vec{\xi}') I(\tau, \vec{\xi}') d\vec{\xi}', \tag{1}$$

where τ is the optical variable, $\vec{\xi} = \vec{\xi}(\mu, \varphi)$ denotes a vector with a polar variation $\mu = \cos\theta$, $(0 \leq \theta \leq \pi)$ and azimuthal variation $0 \leq \varphi \leq 2\pi$, $\beta(\vec{\xi}; \vec{\xi}')$ is the phase function that represents the probability of an incident radiation beam with intensity $I(\tau, \vec{\xi})$, in the $\vec{\xi}$ direction, to be scattered in the $\vec{\xi}' = (\mu', \varphi')$ direction, $d\vec{\xi} = d\mu' d\varphi'$ is the differential element, and finally ϖ is the single scattering albedo.

To complete our formulation of the problem we need to define the boundary conditions associated with the problem, which represent the flux entering and leaving the medium. These conditions are given by

$$I(0, \vec{\xi}) = G_{+}(\vec{\xi}) \quad \text{for} \quad \mu > 0, \tag{2a}$$

and

$$I(\tau_0, \vec{\xi}) = G_{-}(\vec{\xi}) \quad \text{for} \quad \mu < 0. \tag{2b}$$

Taking $\cos\Theta = \langle \vec{\xi}, \vec{\xi}' \rangle$, we can write the phase function as $\beta(\cos\Theta)$. According to Chandrasekhar (1950), we can write the phase function as a finite sum of the Legendre polynomial given in terms of the cosine of the scattering angle Θ, thus

$$\beta(\cos\Theta) = \frac{1}{4\pi} \sum_{l=0}^{L} \omega_l P_l(\cos\Theta), \quad \omega_0 = 1 \quad \text{and} \quad |\omega_l| < 2l + 1 \ \text{ for } \ 0 < l \leq L, \tag{3}$$

where ω_l and P_l are, respectively, the coefficient and the Legendre polynomial in the L^{th}-order expansion of the phase function.

Using the Henyey-Greenstein phase function (Mobley, 1994)

$$\beta(g, \Theta) = \frac{1 - g^2}{\sqrt{(1 - 2g\cos\Theta + g^2)^3}}, \tag{4}$$

where g is a parameter that can be adjusted to control the relative amounts of forward and backward scattering, we obtain the coefficients ω_l (Dall Cortivo, 2008)

$$\omega_n = (2n + 1) g^n, \qquad \text{and} \quad P_0(\cos\Theta) = 1. \tag{5}$$

We can write the intensity as $I(\tau, \mu, \varphi)$ and thus, we can use the Fourier *cosine* decomposition (Chandrasekhar, 1950)

$$I(\tau, \mu, \varphi) = \frac{1}{2} \sum_{m=0}^{L} (2 - \delta_{0,m}) I^m(\tau, \mu) \cos[m(\varphi - \varphi_0)] \tag{6}$$

ISSN 2238-1007

along with the addition theorem (Sansone, 1959) for the Legendre polynomials and express the phase function in the form (Chandrasekhar, 1950)

$$\beta(\cos\Theta) = \frac{1}{4\pi}\sum_{m=0}^{L}(2-\delta_{0,m})\sum_{l=m}^{L}\omega_l P_l^m(\mu')P_l^m(\mu)\cos[m(\varphi'-\varphi)], \tag{7}$$

where

$$P_l^m(\mu) = \left[\frac{(l-m)!}{(l+m)!}\right]^{1/2}(1-\mu^2)^{m/2}\frac{d^m}{d\mu^m}P_l(\mu) \tag{8}$$

denotes an associated Legendre function, to deduce that the original problem can be reduced to a series of $L+1$ uncoupled problems for the coefficients $\{I^m(\tau,\mu)\}$ in Eq. (6). Thus the uncoupled problems can be written for $m = 0,1,\ldots,L$, as

$$\mu\frac{\partial}{\partial\tau}I^m(\tau,\mu) + I^m(\tau,\mu) = \frac{\varpi}{2}\sum_{l=m}^{L}\omega_l P_l^m(\mu)\int_{-1}^{1}P_l^m(\mu')I^m(\tau,\mu')d\mu', \tag{9}$$

subjected to the boundary conditions

$$I^m(0,\mu) = G_+^m(\mu) \quad \text{for} \quad \mu > 0 \tag{10a}$$

and

$$I^m(\tau_0,\mu) = G_-^m(\mu) \quad \text{for} \quad \mu < 0, \tag{10b}$$

and for this problem we consider only the case where $m = 0$.

2.1 The discrete-ordinates solution

To define our discrete-ordinates version of the problem posed by Eqs. (9) and (10), we begin by introducing a quadrature of order N with nodes $\{\mu_j\}$ and weights $\{\eta_j\}$ to approximate the integral in Eq. (9). We set $\mu = \mu_j$, $j = 1,2,\ldots,N$, in Eq. (9) to write the discrete-ordinates equations,

$$\mu_j\frac{d}{d\tau}I(\tau,\mu_j) + I(\tau,\mu_j) = \frac{\varpi}{2}\sum_{l=0}^{L}\omega_l P_l(\mu_j)\sum_{i=1}^{N}\eta_i P_l(\mu_i)I(\tau,\mu_i), \tag{11}$$

subjected to the boundary conditions

$$I(0,\mu_j) = G_+(\mu_j), \tag{12a}$$

and

$$I(\tau_0,\mu_j) = G_-(\mu_j), \tag{12b}$$

where we assume that the nodes of the quadrature scheme are ordered in such a way that the first n ($n = N/2$) nodes are *positive* and the others $N-n$ nodes are *negative*, noting that $\mu_j = -\mu_{n+j}$ and $\eta_j = \eta_{n+j}$, for $j = 1,2,\ldots,n$.

Now we assume that $I(\tau,\mu_j)$ can be written as

$$\vec{I}_+(\tau) = [I(\tau,\mu_1), I(\tau,\mu_2),\ldots,I(\tau,\mu_n)]^T \tag{13a}$$

and

$$\vec{I}_-(\tau) = [I(\tau,\mu_{n+1}), I(\tau,\mu_{n+2}),\ldots,I(\tau,\mu_N)]^T, \tag{13b}$$

where $\vec{I}_+(\tau)$ only contains the values associated with $\mu > 0$, $\vec{I}_-(\tau)$ only contains the values associated with $\mu < 0$ and T denotes the transpose, we then obtain

$$\vec{I}(\tau) = \begin{bmatrix} \vec{I}_+(\tau) \\ \vec{I}_-(\tau) \end{bmatrix}. \tag{14}$$

So we can write Eq. (11) in an matrix form

$$\frac{d}{d\tau}\vec{I}(\tau) = \mathbb{A}\vec{I}(\tau), \tag{15}$$

where each entry of matrix $\mathbb{A}$ is given by

$$a_{ij} = -\frac{\delta_{ij}}{\mu_j} + \frac{\varpi}{2\mu_j}\sum_{l=0}^{L}\omega_l P_l(\mu_j)P_l(\mu_i)\eta_i, \qquad i,j = 1,2,\ldots,N, \tag{16}$$

ISSN 2238-1007

with $\delta_{ij} = 1$ if $i = j$, $\delta_{ij} = 0$ if $i \neq j$, and the boundary conditions given by

$$\vec{I}(0) = \vec{G}_+(\mu_j), \qquad \text{if} \qquad 1 \leq j \leq N/2 \tag{17a}$$

and

$$\vec{I}(\tau_0) = \vec{G}_-(\mu_j), \qquad \text{if} \qquad N/2+1 \leq j \leq N. \tag{17b}$$

To solve the system expressed by Eq. (15) we consider some proprieties of matrix $\mathbb{A}$. According to Case & Zweifel (1967), matrix $\mathbb{A}$ has a complete set of eigenvalues and eigenvectors (under some considerations of ϖ parameter), where half of these eigenvalues are positive and the other half are negative. So matrix $\mathbb{A}$ can be diagonalized as

$$\mathbb{A} = \mathbb{P}\mathbb{D}\mathbb{P}^{-1}, \tag{18}$$

where $\mathbb{P}$ is a matrix containing the eigenvectors of $\mathbb{A}$, $\mathbb{D}$ is a diagonal matrix containing the eigenvalues of $\mathbb{A}$, and $\mathbb{P}^{-1}$ is the inverse matrix of the eigenvectors. Now if we consider $\vec{I}(\tau) = \mathbb{P}\vec{y}(\tau)$ we can write Eq. (15) as

$$\frac{d}{d\tau}\vec{y}(\tau) = \mathbb{D}\vec{y}(\tau), \tag{19}$$

whose solution is given by

$$\vec{y}(\tau) = e^{\mathbb{D}\tau}\vec{k}. \tag{20}$$

From $\vec{I}(\tau) = \mathbb{P}\vec{y}(\tau)$ and the solution expressed by Eq. (20) we can write the final solution for the system given by Eq. (15) as

$$\vec{I}(\tau) = \mathbb{P}e^{\mathbb{D}\tau}\vec{k}. \tag{21}$$

As half of the eigenvalues of matrix $\mathbb{A}$ are positive and the other half are negative, and after assorting them in decreasing order, we define the negative eigenvalues as matrix $\mathbb{D}^-$ and the positive eigenvalues as matrix $\mathbb{D}^+$, thus matrix $\mathbb{D}$ can be written as

$$\mathbb{D} = \begin{bmatrix} \mathbb{D}^- & 0 \\ 0 & \mathbb{D}^+ \end{bmatrix}. \tag{22}$$

And now, the matrix of the eigenvectors can be written as

$$\mathbb{P} = \begin{bmatrix} \mathbb{P}_{11} & \mathbb{P}_{12} \\ \mathbb{P}_{21} & \mathbb{P}_{22} \end{bmatrix}, \tag{23}$$

where $\mathbb{P}_{ij}$ is a representation for each block of the eigenvectors matrix.

Note that the solution given by Eq. (21) contains the direction $\mu > 0$ and $\mu < 0$. According to Duderstadt & Martin (1975) we can separate Eq. (21) into two solutions, *i.e.*, a solution containing only the directions associated with $\mu > 0$ and the other solution the directions associated with $\mu < 0$, thus we obtain, respectively,

$$\begin{bmatrix} \vec{I}_+(\tau) \\ \vec{I}_+(\tau) \end{bmatrix} = \begin{bmatrix} \mathbb{P}_{11} & \mathbb{P}_{12} \\ \mathbb{P}_{21} & \mathbb{P}_{22} \end{bmatrix} \begin{bmatrix} e^{\mathbb{D}^-\tau} & 0 \\ 0 & e^{\mathbb{D}^+(\tau-\tau_0)} \end{bmatrix} \begin{bmatrix} \vec{k}_1 \\ \vec{k}_2 \end{bmatrix} \tag{24}$$

and

$$\begin{bmatrix} \vec{I}_-(\tau) \\ \vec{I}_-(\tau) \end{bmatrix} = \begin{bmatrix} \mathbb{P}_{11} & \mathbb{P}_{12} \\ \mathbb{P}_{21} & \mathbb{P}_{22} \end{bmatrix} \begin{bmatrix} e^{\mathbb{D}^-\tau} & 0 \\ 0 & e^{\mathbb{D}^+(\tau-\tau_0)} \end{bmatrix} \begin{bmatrix} \vec{k}_1 \\ \vec{k}_2 \end{bmatrix}. \tag{25}$$

To determine the values of the vectors $\vec{k}_1$ and $\vec{k}_2$ we use the boundary conditions. From Eqs. (17), as we know the radiation intensity at $\tau = 0$ for $\mu > 0$ and $\tau = \tau_0$ for $\mu < 0$, therefore, we take the first line of the system expressed by Eq. (24) to obtain

$$\vec{I}_+(\tau) = \mathbb{P}_{11}e^{\mathbb{D}^-\tau}\vec{k}_1 + \mathbb{P}_{12}e^{\mathbb{D}^+(\tau-\tau_0)}\vec{k}_2 \tag{26}$$

and take the second line of the system expressed by Eq. (25) to obtain

$$\vec{I}_-(\tau) = \mathbb{P}_{21}e^{\mathbb{D}^-\tau}\vec{k}_1 + \mathbb{P}_{22}e^{\mathbb{D}^+(\tau-\tau_0)}\vec{k}_2. \tag{27}$$

Now, we apply to the above equations the boundary condition in Eqs. (17). To the equation associated with $\vec{I}_+(\tau)$ we apply Eq. (17a), and to the equation associated with $\vec{I}_-(\tau)$ we apply Eq. (17b), thus we obtain

$$\begin{bmatrix} \vec{I}_+(0) \\ \vec{I}_-(\tau_0) \end{bmatrix} = \begin{bmatrix} \vec{G}_+(\mu_j) \\ \vec{G}_-(\mu_j) \end{bmatrix} = \begin{bmatrix} \mathbb{P}_{11} & \mathbb{P}_{12}e^{-\mathbb{D}^+(\tau_0)} \\ \mathbb{P}_{21}e^{\mathbb{D}^-\tau_0} & \mathbb{P}_{22} \end{bmatrix} \begin{bmatrix} \vec{k}_1 \\ \vec{k}_2 \end{bmatrix}. \tag{28}$$

Once the linear system formulated by Eq. (28) is solved for $\vec{k}_1$ and $\vec{k}_2$, we have at hand all quantities necessary to evaluate Eqs. (24) and (25) for any $\tau \in [0, \tau_0]$.

ISSN 2238-1007

3 METHODOLOGIES

We compare the results presented in this work with those shown in Cortivo et al. (2010). For practical purposes, we refer to results presented in Cortivo et al. (2010) as Methodology-1 utilizing a neural network ANN-1 and the results presented in this paper as Methodology-2 utilizing a neural network ANN-2.

3.1 Training and Validation Sets

The training and validation sets, which are the same for both methodologies, were obtained by the numerical solution of the problem summarized in Section 2 that determines, for the analyzed layer of water, the emitted radiation at the surface, $\tau = 0$, for the selected discrete negative directions, $\mu < 0$, and the emitted radiation at the bottom, $\tau = \tau_0$, for the selected discrete positive directions, $\mu > 0$. The number of network inputs is associated with the discrete directions and the number of patterns is associated with the discrete values of the single scattering albedo ϖ, as each discrete value of ϖ generates a direct problem, and consequently generates a pattern of training/validation set. The difference between these sets is due to the type of discretization applied to the single scattering albedo, as described in the following paragraphs.

The emitted radiation (radiances) were computed by using the following parameters (see Section 2): $\varpi \in [0.2, 0.9]$, $\mu_0 = 0.5$ (an incident polar angle $\theta_0 = 60^o$), $\varphi_0 = 0^o$, $G_+ = 1, G_- = 0$ and $N = L = 150$. Although the acceptable physical values for $\varpi \in (0,1)$, we set the interval to $\varpi \in [0.2, 0.9]$, in order to adequately represents the types of natural waters: clear ocean, coastal ocean and turbid harbor, that are utilized in hydrologic optic problems (Mobley, 1994). With $N = 150$, we were able to generate, for the polar angle variable, 150 discrete directions: 75 positive directions ($\mu > 0$) and 75 negative directions ($\mu < 0$).

For the training and validation sets, we only selected 10 values from the 75 generated radiances with $\mu < 0$, that are associated with 10 equally spaced values of $\theta \in (90^o, 151.2^o]$ to compose half of the network inputs. To the other half, we attributed 10 values of G_+, associated with $\mu > 0$. Finally, to compose all the patterns for the training set we discretized the single scattering albedo with a step size of $\Delta_\varpi = 0.01$, thus generating 71 values of

$$\varpi = [0.2, 0.21, \ldots, 0.89, 0.90]. \tag{29}$$

So, the training set was composed of 71 patterns with 20 inputs each, and each pattern was associated with a direct problem solution. The validation set was also composed of 20 inputs, however the total number of patterns was extended to 7001, as we used a much smaller step size $\Delta_\varpi = 0.0001$ in the discretization process, noting that 71 of these 7001 patterns were the same used in the training set. The justification for the addition of 10 inputs associated with the directions $\mu > 0\,(G_+)$, is due to the fact that each pattern of the set takes a similar form of the activation function used in the hidden layer, as we will see in the next subsection. Figure 1 presents the training set used in Methodology-1.

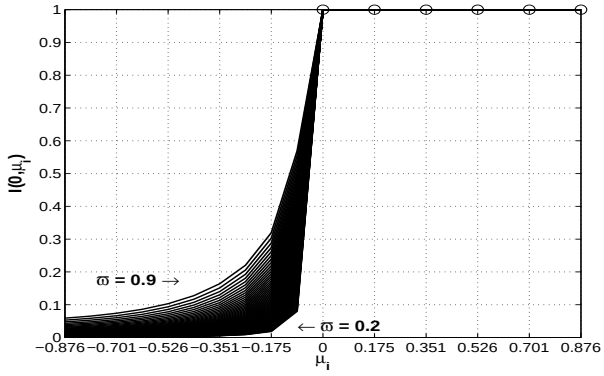


Figure 1: Graphical representation of the training set used in Methodology-1.

3.2 Methodology-1 with ANN-1

The network topology used in Methodology-1 consisted of 20 input neurons, which represented the size of the input vector, one hidden layer with 40 neurons and one output layer with only one neuron. We used in the output layer a linear function as the activation function

$$\phi(v) = av \qquad \text{with} \qquad a = 1, \tag{30}$$

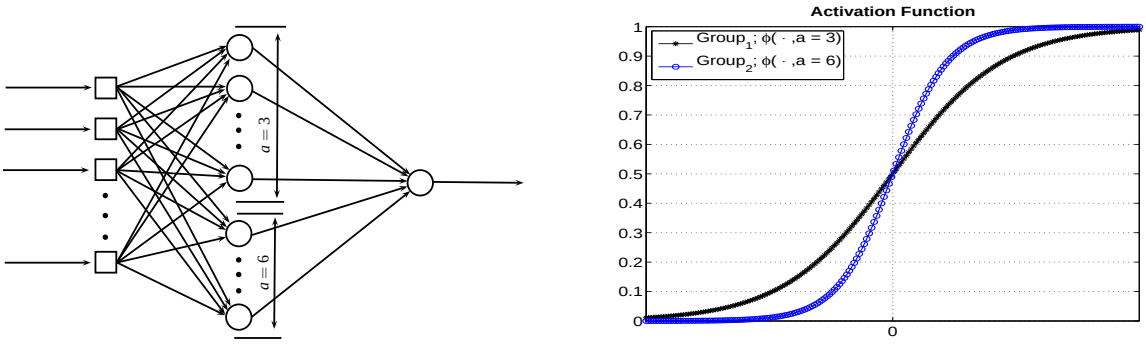
and in the hidden layer a sigmoid function

$$\phi(v) = \frac{1}{1 + e^{-av}}, \tag{31}$$

where the parameter $a = 3$ for the first 30 neurons (Group 1) and $a = 6$ for the last 10 neurons (Group 2). Figure 2(a) shows the topology of the network used and Fig. 2(b) illustrates the two activation functions used in the hidden layer.

The justification for the use of two activation functions in the hidden layer is due to the behavior of the pattern sets. In Fig. 1, we note that the radiation curves generated with ϖ values close to 0.2 are almost equivalent to zero for almost

ISSN 2238-1007



(a) Graphical representation of the network topology.

(b) Graphical representation of the activation functions used in the hidden layer.

Figure 2: Graphical representation of the network topology and activation functions used in the hidden layer of Methodology-1.

all the range of $\mu < 0$, and as ϖ values increase, these curves tend to move away from zero. Therefore, as the activation function with $a = 3$ is less steeper than the one with $a = 6$, the idea is to make the neurons that contain the activation function with $a = 3$ specialize in recognizing the radiances generated with ϖ values close to 0.9 and let the other neurons specialize in others radiances generated with ϖ values close to zero.

In addition, we used the backpropagation algorithm error with the update of the synaptic weights, based on the generalized delta rule, defined by (Haykin, 1999)

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \alpha\left[w_{ji}^{(l)}(n-1)\right] + \eta\,\delta_j^{(l)}(n) y_i^{(l-1)}(n), \tag{32}$$

where l represents the layer, $w_{ji}^{(l)}$ is the synaptic weight of neuron j in layer l that is fed from the neuron i of layer $l-1$, $\delta_j^{(l)}$ is the local gradient of neuron j in the layer l, $y_i^{(l-1)}(n)$ is the output signal of neuron i in the previous layer $l-1$ at iteration n, η is the learning rate and α is the momentum constant.

During the training experiments, we observed that the training should be performed in four stages, and each stage was composed of 100000 epochs. For the first two stages, we found out that 71 patterns were not sufficient for reaching a good network learning, so we opted for the increase of the number of patterns. We then created two more sets of identical patterns as described before and added these two set in the previous training set, therefore our new training set was composed of 213 patterns (vectors) with 20 positions (inputs) each, that were randomized and then separated into four groups, and then noise levels were added to them, as shown in Tab. 1. The addition of noise was necessary due to the fact that these data were not observed data, but synthetic data generated by a computer.

Table 1: Pattern sets for training the neural network in the first two stages

Groups	1	2	3	4
Patterns range	1–34	35–71	72–142	143–213
Noise	0%	10%	5%	2%
Patterns percentage	16%	17%	33%	34%

For the last two stages of training, we used the original 71 patterns, which were also randomized and divided into four groups, and then the same noise levels were added to each set as shown in Tab. 2.

Table 2: Pattern sets for training the neural network in the last two stages

Groups	1	2	3	4
Patterns range	1–10	11–21	22–46	47–71
Noise	0%	10%	5%	2%
Patterns percentage	16%	17%	33%	34%

Table 3 shows the variation of the training rate η and the momentum constant α for each training stage. The number of epochs, the learning rate and the momentum values were determined through various training experiments, which is the normal procedure used in the network training process.

Table 3: Values of α and η used in the training in each epoch interval for all stages

Epochs	1–5000	5001–20000	20001–30000	30001–35000	35001–70000	70001–100000
η	0.01	0.009	0.009	0.0005	0.0001	0.000005
α	0.1	0.4	0.08	0.009	0.001	0.0005

For the first training stage, we used a random number generator to obtain the initial weights w, which were divided by 10^9 in order to try to insert a minimal knowledge in the neural network. The second stage consisted of a new training,

ISSN 2238-1007

however, for the weights and biases, we used the initial values returned from the first training. Note that in this case, we started the training process from a possible local minimum and, from this point on, we tried to obtain another local minimum by changing the values of η and α. This technique was also adopted for the last two stages, where we considered the data that were presented in Tab. 2. Figure 3 exemplifies the four training stages process.

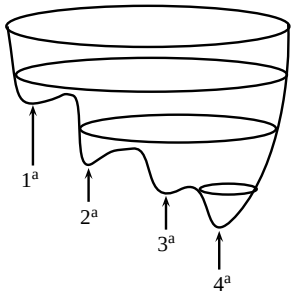


Figure 3: Graphical representation of the network training process with backpropagation.

We adopted this strategic training after performing many experiments, and we justify the use of four stages of training with the following arguments. At the end of each training the validation set was presented to the network five times, as follows: at first the validation set was presented without noise; in the second the set was corrupted with 2% of noise; in the third and fourth the set was presented with 5% and 10% of noise, respectively, and finally in the fifth it was presented with 7% of noise. Note that the training was composed of 213 patterns (first two stages) and 71 patterns (last two stages) and four levels of noise (0%, 2%, 5% e 10%), whilst, the validation set was composed of 7001 patterns, including an additional noise level that was not present in the training. The reason for that was to try to generalize the capacity of the network to recognize patterns not present in the learning process.

Although it was a long network training, there was not a possibility for the occurrence of *overfitting*, as it was justified in Cortivo et al. (2010). Finally, Tab. 4 shows the correct answers (hits rate) obtained from the validation set for each noise level considered in the training process and additional noise, only present in the validation set.

Table 4: Hits rate obtained with Methodology-1.

Noise levels	0.00%	2.00%	5.00%	10.00%	7.00%
Hits rate	100.00%	94.12%	94.60%	95.39%	94.93%

3.3 Methodology-2 with ANN-2

In this methodology, we describe how it was formulated the optimization problem for the training technique adopted in this work. Here we use the quasi-Newton method, which is also a supervised type algorithm. We begin by assuming that each training pattern is associated with a vector $\vec{I}_i$, where i is, in its turn, associated with a discrete value of ϖ. By organizing all the vectors $\vec{I}_i$ in a matrix form $\mathbb{E}_t$ (training input), we have

$$\mathbb{E}_t = \left[\vec{I}_1, \vec{I}_2, \ldots, \vec{I}_T\right], \tag{33}$$

where $T = 71$ is the total number of training patterns. By performing the same for the validation set, where $\mathbb{E}_v$ (validation input) is the matrix, we have

$$\mathbb{E}_v = \left[\vec{I}_1, \vec{I}_2, \ldots, \vec{I}_V\right], \tag{34}$$

where $V = 7001$ is the total number of validation patterns. Note that each matrix column is a solution (emitted radiation) of a direct problem, generated by a determined ϖ value, will be used as input of the network (inverse problem).

By organizing all discretized ϖ values in a vector, in the same order of the above mentioned matrix columns, we obtain a vector $\vec{S}_t^e$ (exact output)

$$\vec{S}_t^e = [\varpi_1, \varpi_2, \ldots, \varpi_T], \tag{35}$$

where ϖ_i follows the discretization process given in Eq. (29). Note that ϖ values were the input parameters to the direct problem, then it became the network output.

The proposed training method needs the weight $\mathbb{W}$ and the biases $\mathbb{B}$ matrices, to be written as a vector $\vec{X}$, which contains all network free variables, thus

$$\vec{X} = [\mathbb{W}, \mathbb{B}]. \tag{36}$$

We associated $\vec{S}^r$ (network output) with the calculated output of the network. Observe that the network output depends on each training pattern, $\vec{I}_i$, and of the vector $\vec{X}$ that contains the weights and biases, thus by Eq. (36) we have

$$\vec{S}^r = \vec{S}^r\left(\vec{I}_i, \vec{X}\right). \tag{37}$$

ISSN 2238-1007

Now, we formulate the optimization problem (training) with restrictions based on the functional minimization

$$\mathscr{J}\left(\mathbb{E}_t, \vec{X}\right) = \min \sum_{i=1}^{T} \left|\left| \vec{S}^e_{t_i} - \vec{S}^r\left(\vec{I}_i, \vec{X}\right) \right|\right|_2^2, \tag{38}$$

subjected to the following restrictions

$$\begin{array}{rcccl} -w^k_{\min} & \leq & w^k & \leq & w^k_{\max} \\ -b^l_{\min} & \leq & b^l & \leq & b^l_{\max} \end{array}, \tag{39}$$

where w^* e b^* are contained in vector $\vec{X}$, and $\vec{S}^e_{t_i}$ represents each one of the inputs present in Eq. (35). To minimize the above functional, Eq. (38), we utilized subroutine E04UCF, implemented in the NAG Library (NAG, 1995). The training in this formulation was done in batch mode, *i.e.*, all training patterns were presented to the network, then it was performed the update of the network variables: $\mathbb{W}$ e $\mathbb{B}$. The update of these variables, performed by subroutine E04UCF can be explained by the following paragraph.

During the network training, subroutine E04UCF search for a set of variables $\vec{X}^*$ for $\vec{X}$, Eq. (36), such that

$$\nabla \mathscr{J}\left(\cdot, \vec{X}^*\right) = \vec{0}, \tag{40}$$

where, $\vec{X}^*$ must satisfy the conditions given by Eq. (39). More details for solving the above equation can be found in NAG (1995) Manual.

3.3.1 Training and Validation

Initially, we used topology ANN-1 with the proposed method (quasi-Newton), but during the network training process we noticed that some modifications could be performed in the network topology: the reduction of the number of the network input and the number of neurons in the hidden layer, which resulted in the creation of ANN-2 topology. As the proposed method presented better results with the ANN-2 topology and for practical purposes, we will omit all results associated with ANN-1 topology.

Regarding the number of inputs of each training pattern, which are associated with the network input, we noticed that only the values associated with the emitted radiances at $\tau = 0$ and with directions $\mu < 0$ would be sufficient, so we modified the number of network inputs from 20 to 10. Moreover, it was possible to change the number of neurons in the hidden layer from 40 to 20, maintaining the same proportion in each group of neurons, as in Methodology-1, *i.e.*, in the first group which has the parameter of activation function $a = 3$ we put 15 neurons, and in the second group with $a = 6$ we put five neurons. See Figs. 2(a) and 2(b) for the network topology and activation functions, respectively. Observe that the activation functions used in Methodology-2 are the same used in Methodology-1, and the network topology is also the same, except for the number of neurons in the input and the hidden layers.

Figure 4 shows graphically the pattern set for the training used in Methodology-2. Another modification was that we only used one stage with 71 patterns for the network training, instead of using four stages, with 213 patterns in the first two and 71 patterns in the last two, as in Methodology-1 (Cortivo et al., 2010). For the validation set we adopted the same criteria of the training set, *i.e.*, we excluded the directions (network input) associated with $\mu > 0$ values at $\tau = 0$, thus the validation set was also composed of 10 inputs, but we maintained the 7001 patterns, as before.

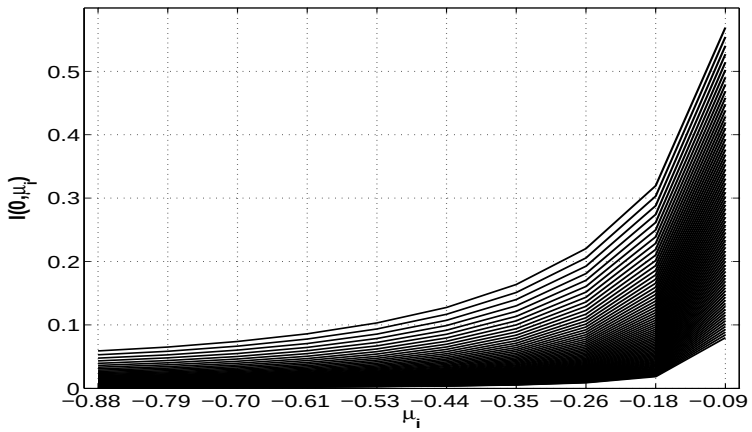


Figure 4: Graphical representation of the training set used in Methodology-2

The neural network training is as follows: firstly, the 71 patterns were placed in random order, then separated into four groups and finally corrupted with the same noise levels shown in Tab. 2. For the computational procedure of the training we created a FORTRAN 90 routine called OBJFUN which implemented the artificial neural network and the defined functional in Eq. (38). This routine works together with other routines present in NAG library (NAG, 1995), which contains criteria to search the “optimal” solution through the quasi-Newton method.

ISSN 2238-1007

In this method, one of the stopping criteria defined by the user is the iterations number (iterations number is equivalent to the number of epochs used in the backpropagation algorithm) as well as other criteria which are defined and controlled by the NAG library. As we do not know the ideal iterations number we adopted the following procedure. An *if-condition* was created in OBJFUN routine to perform the control of the Cross Validation process, *i.e.*, to test the performance (generalization) of the network during the training with the validation set. Every time the validation set is presented to the network, all information, such as, the network variables values, the total relative error, the correct answers in each case (noise level), are stored. When a new call occurs, the validation set is presented again to the network, and the new information are compared with the old ones and, in case its necessary, the information that generated the best answers are stored by replacing the old ones. At the end of the iterations number (epochs) defined by the user, or NAG criteria, the information that presented the best generalization are recovered. Each time the training is paused by the *if-condition*, we presented to the network the 7001 patterns of the corrupted set with the same noise levels of the training, in addition to the noise level of 7%, referenced before.

4 RESULTS

For the sake of clarity, we initially explain some important points:

- We only discuss two methodologies: Methodology-1 (Cortivo et al., 2010) and Methodology-2 (this work), and the difference between them is the learning process;
- Two networks, the ANN-1 and the ANN-2, associated with Methodology-1 and Methodology-2, respectively, were trained;
- The ANN-1 was trained with the backpropagation algorithm error, and the ANN-2 was trained with the quasi-Newton method;
- Each one of these networks was capable of solving 7001 problems in hydrologic optics, that were corrupted with five levels of noise.

In this section we present the obtained results with the proposed training methodology (Methodology-2), and the results previously obtained using Methodology-1 (Cortivo et al., 2010). Here, we present the single scattering albedo estimation and the relative error graphics.

In the counting process, we only considered correct answers when $\varpi - R \cdot \varpi \leq \varpi^r \leq \varpi + R \cdot \varpi$, where ϖ^r is the estimated albedo by the network and R is the percentage of noise level. For the case of 0% of noise, we considered a tolerance of $\varepsilon = \pm 0.005$ for the exact value. Table 5 shows the correct answers (hits) given by ANN-2 applied to the validation set for each considered noise level. As can be noted, there was a 100% of correct answers in all cases, which differs from Methodology-1 where it was obtained 100% of correct answer only for noiseless input data.

Table 5: Hits obtained with Methodology-2.

Noise levels	0.00%	2.00%	5.00%	10.00%	7.00%
Hits	100.00%	100.00%	100.00%	100.00%	100.00%

Figures 5, 7, 9, 11 and 13 show the single scattering albedo ϖ estimated by the networks. The red and the black lines represent the exact and the estimated values of ϖ, respectively. The blue lines (dash and dot) represent

$$\pm \text{Exa} = \varpi \pm R \cdot \varpi, \tag{41}$$

where Exa is the blue line ($+$Exa above the red line and $-$Exa below the red line). For the data with 0% of noise, we adopted

$$\pm \text{Exa} = \varpi \pm 0.005. \tag{42}$$

For a better visualization of the results, each figure was divided into two graphs.

Figures 6, 8, 10, 12 and 14 show the relative error obtained by the networks

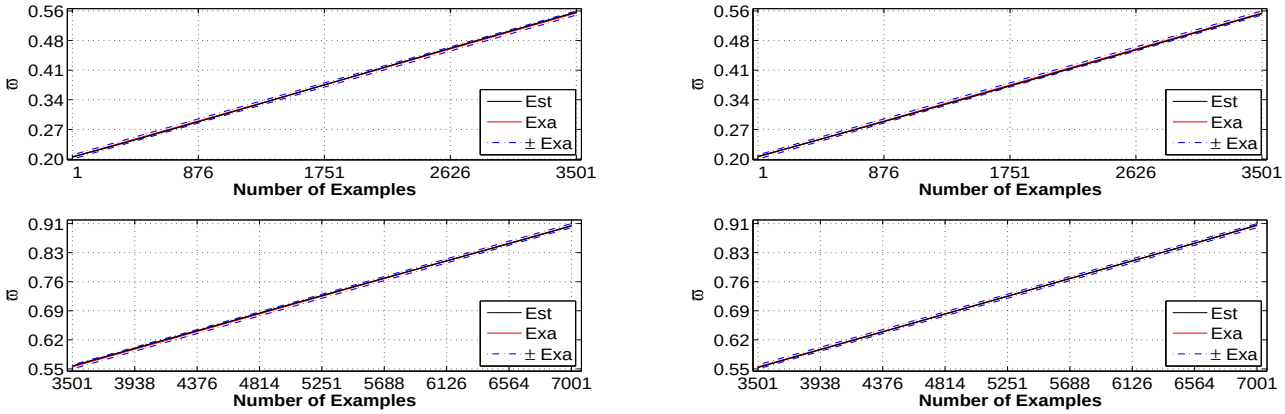
$$\varepsilon = \frac{\varpi^r - \varpi}{\varpi} \times 100. \tag{43}$$

The blue lines (dash and dot) represent the boundary lines ($\pm R$) for the noise levels, and in Fig. 6 we considered an interval of $\pm 1\%$, as the input data were not corrupted with noise.

Figure 5 shows the albedos recovered when presented to the network the noiseless training set. Note that the black lines (estimated albedos) are within the defined interval by the blue lines in both methodologies, which is due to the fact that both networks have presented a 100% of correct answers.

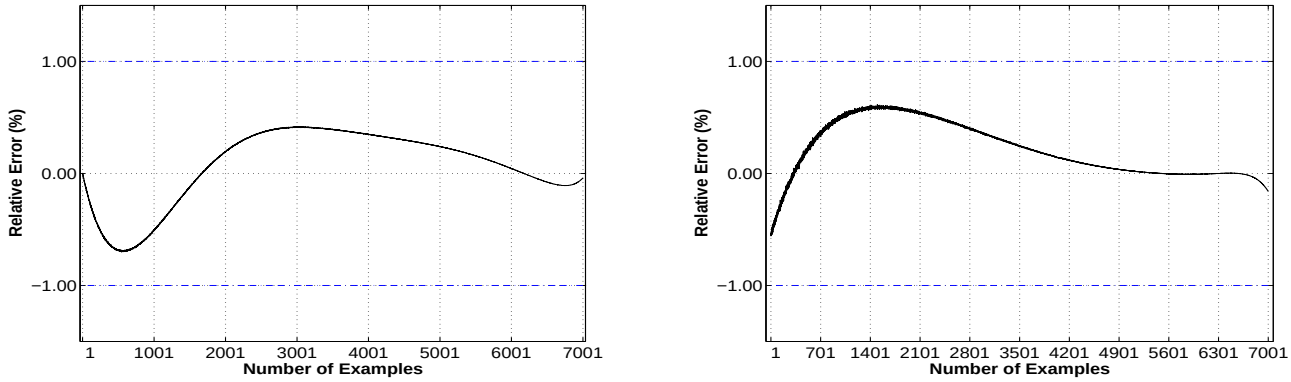
However, in comparing Tabs. 4 and 5, which show the correct answers of each network, we expected that, in the figures related to Methodology-1, the ϖ values would not be completely contained within the defined area by the blue lines, which is contrary to what was expected from Methodology-2.

ISSN 2238-1007



(a) Single scattering albedo estimated by ANN-1 (b) Single scattering albedo estimated by ANN-2

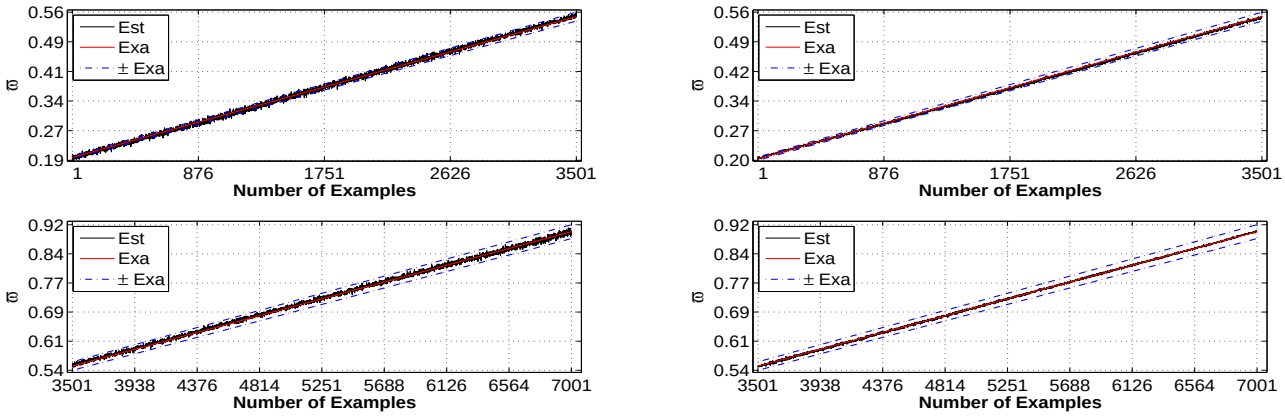
Figure 5: Comparisons between the results obtained by the neural networks for noiseless input data



(a) Relative error obtained by ANN-1 (b) Relative error obtained by ANN-2

Figure 6: Relative error obtained by the neural networks for noiseless input data

Figure 6 presents the relative errors of each estimation that are shown in Fig. 5. We can also observe from Fig. 6, that the relative error is less than 1%, which is acceptable because the input data were not corrupted with noise. We also observe that the oscillation of the error curve in Fig. 6(a) is greater than the one shown in Fig. 6(b).



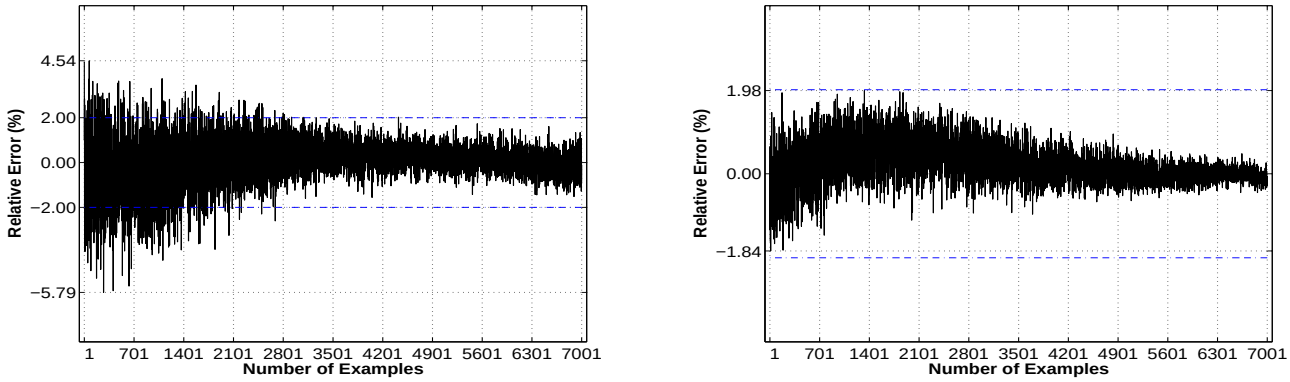
(a) Single scattering albedo estimated by ANN-1 (b) Single scattering albedo estimated by ANN-2

Figure 7: Comparisons between the results obtained by the neural networks for input data with 2% of noise

As we will see below, this oscillation present in the first examples (problems 1-3000), which is equivalent to $\varpi \in [0.2, 0.5]$, will also be present in the other analyzed cases. This is due to the fact that the radiation curves are very close to each other for distinct albedo values, which presents a difficulty for a correct classification by the network. Note that, although the second training methodology is more robust, these oscillation will still remain in the results, yet in a controlled form, *i.e.*, the oscillations are contained within the interval of a pre-defined tolerance error, in all considered cases even with data corrupted with 7% of noise.

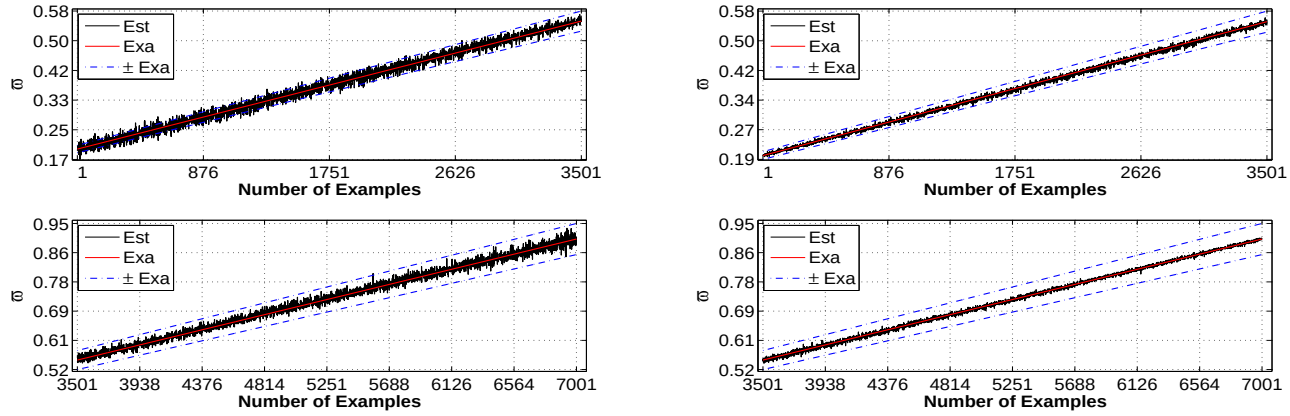
ISSN 2238-1007

Below, we present the other figures with the estimated albedo and the respective relative errors. The figures on the left side represent the results obtained by ANN-1 and those on the right side by ANN-2. From these figures, we clearly note that the results obtained from Methodology-2 are much better than those obtained from Methodology-1.



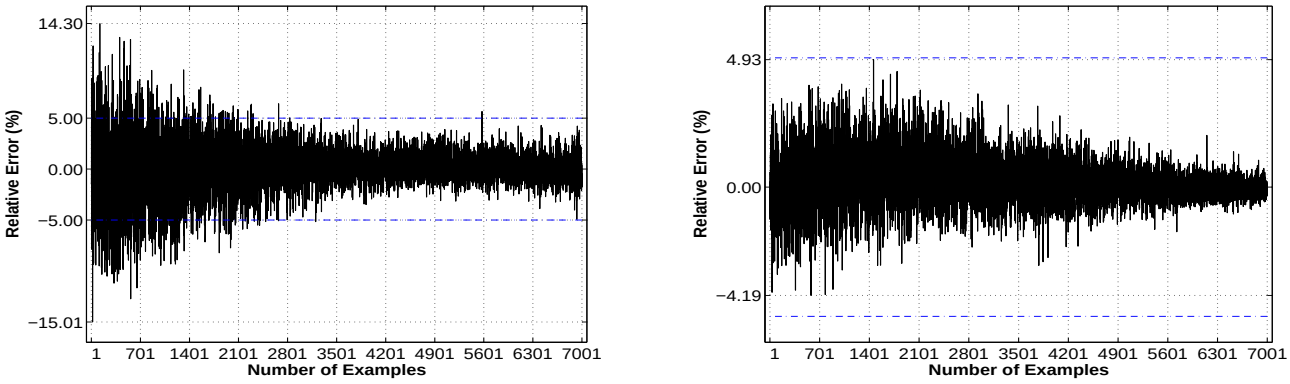
(a) Relative error obtained by ANN-1 (b) Relative error obtained by ANN-2

Figure 8: Relative error obtained by the neural networks for input data with 2% of noise



(a) Single scattering albedo estimated by ANN-1 (b) Single scattering albedo estimated by ANN-2

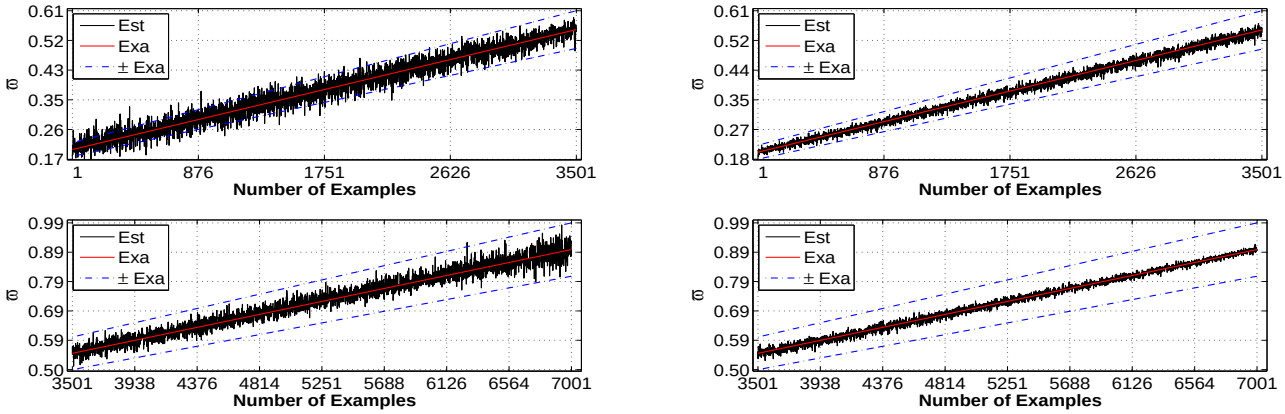
Figure 9: Comparisons between the results obtained by the neural networks for input data with 5% of noise



(a) Relative error obtained by ANN-1 (b) Relative error obtained by ANN-2

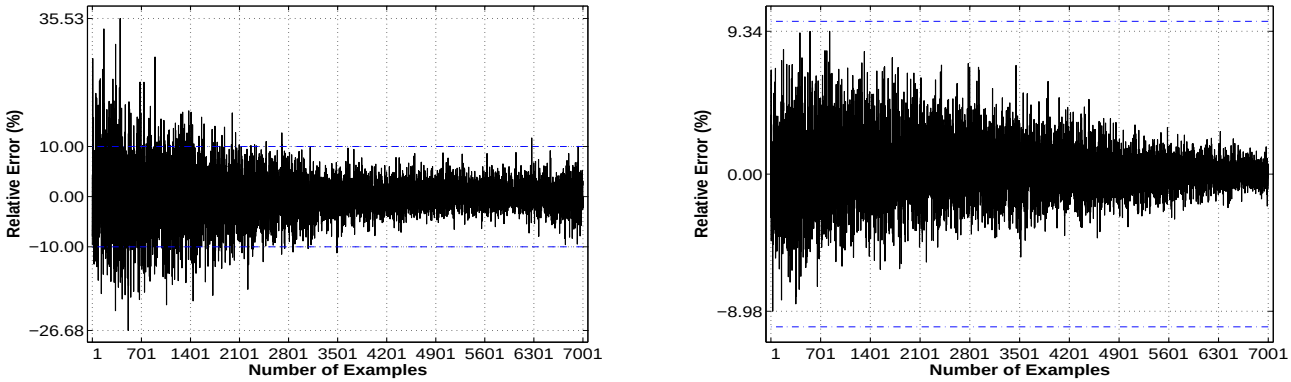
Figure 10: Relative error obtained by the neural networks for input data with 5% of noise

ISSN 2238-1007



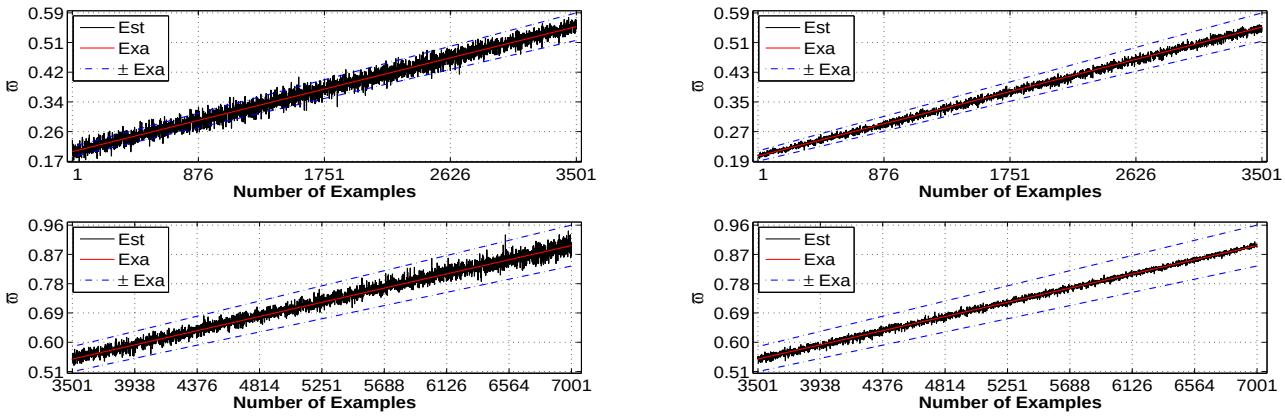
(a) Single scattering albedo estimated by ANN-1 (b) Single scattering albedo estimated by ANN-2

Figure 11: Comparisons between the results obtained by the neural networks for input data with 10% of noise



(a) Relative error obtained by ANN-1 (b) Relative error obtained by ANN-2

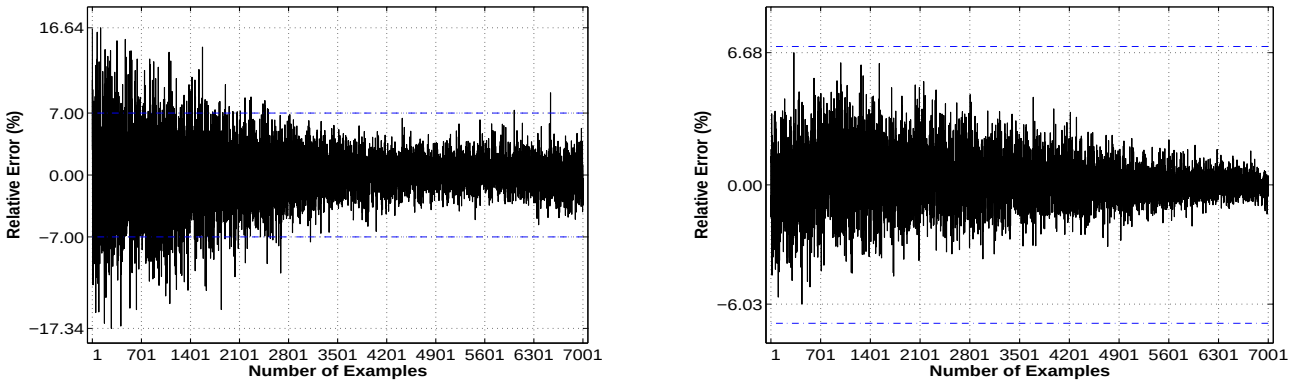
Figure 12: Relative error obtained by the neural networks for input data with 10% of noise



(a) Single scattering albedo estimated by ANN-1 (b) Single scattering albedo estimated by ANN-2

Figure 13: Comparisons between the results obtained by the neural networks for input data with 7% of noise

ISSN 2238-1007



(a) Relative error obtained by ANN-1

(b) Relative error obtained by ANN-2

Figure 14: Relative error obtained by the neural networks for input data with 7% of noise

We present now an analysis regarding the computational time involved in the networks training. An important fact is that in the training with backpropagation we used four stages with 10^5 epochs each, and the training was the *online* type, *i.e.*, each time the pattern is presented to the network, an update is done on the variables of the network. For the first two stages, with 213 patterns and 10^5 epochs in each training, the processing CPU time was 462 s, and for the last two stages, with 71 patterns and 10^5 epochs in each training, the processing CPU time was 144 s. Thus the total CPU time spent in the training with backpropagation was 606 s, in comparison to the CPU time of 10 s involved in the training with quasi-Newton method. The processing was performed in a notebook Dell Vostro 1520 with Intel® processor, Core™ 2 Duo P8600 2.40GHz L2 3Mb, 4Gb of RAM memory, OS GNU/Linux 64 bits (Ubuntu 11.04), Intel® Fortran Composer XE compiler (Build option = -O3).

In Fig. 15 we show the learning curve for the network ANN-1 and ANN-2, in log-log scale. We note that, while the quasi-Newton error curve shows a slow response in the beginning of the training then it converges quickly to a global/local minimum, the Backpropagation error curve takes a longer time in the first two stages, then it afterwards converges after the last two stages.

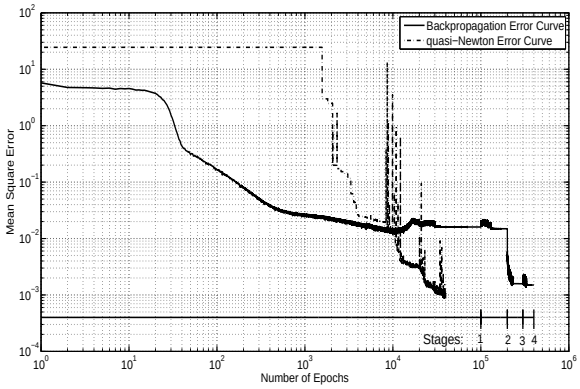


Figure 15: Learning curve of the network ANN-1 and ANN-2.

5 CONCLUSIONS

We compare the two adopted techniques, Methodology-1 and Methodology-2, regarding the following qualities:

- Robustness
  1. In the comparison between these two techniques, we clearly observe the robustness of the quasi-Newton method, when used to solve these type of problems.
  2. Methodology-2 considered the network training as an optimization problem with restrictions in the weights and biases, thus avoided the saturation of neurons. This advantage did not exist in Methodology-1.
- Efficiency
  1. The training with Methodology-2 was faster, simple and more efficient when compared with the training adopted by Methodology-1.
  2. Only with Methodology-2, we were able to reduce the number of inputs, by eliminating the inputs of the training vectors associated with the positive directions, $\mu > 0$ at $\tau = 0$, and also to reduce by 50% the number of neurons in the hidden layer.

ISSN 2238-1007

- Precision

1. In all problems solved, the network trained with the quasi-Newton method obtained better results when comparing with the backpropagation method. Also observe that the recovered albedos, above 0.5 values, presented a little oscillation around the exact values when using the proposed techniques (Methodology-2), in contrast to the large oscillation present in the results obtained by Methodology-1.
2. Methodology-2 presented, in all noise levels, a 100% correct answers (hits) and a relative error smaller than the noise level that corrupted the input data. This precision in the results was not present with Methodology-1.

6 ACKNOWLEDGEMENTS

FDC wishes to thank CAPES for the financial aid granted.

REFERENCES

Bokar, J. C. (1999). The estimation of spatially varying albedo and optical thickness in a radiating slab using artificial neural networks. *International Communications in Heat and Mass Transfer*, *26*, 359 – 367.

Case, K. M., & Zweifel, P. F. (1967). *Linear transport theory*. USA: Addison-Wesley.

Chalhoub, E. S. (1997). O método das ordenadas discretas na solução da equação de transporte em geometria plana com dependência azimutal.

Chalhoub, E. S., & Campos Velho, H. F. (2001). Simultaneous estimation of radiation phase function and albedo in natural waters. *Journal of Quantitative Spectroscopy & Radiative Transfer*, *69*, 137 – 149.

Chalhoub, E. S., & Campos Velho, H. F. (2002). Estimation of the optical properties of sea water from measurements of exit radiance. *Journal of Quantitative Spectroscopy & Radiative Transfer*, *72*, 551 – 565.

Chalhoub, E. S., & Campos Velho, H. F. (2003). Multispectral reconstruction of bioluminescence term in natural waters. *Applied Numerical Mathematics*, *47*, 365 – 376.

Chalhoub, E. S., Campos Velho, H. F., Ramos, F. M., & Claeyssen, J. C. R. (2000). Phase function estimation in natural waters using discrete ordinates method and maximum entropy principle. *Hybrid Methods in Engineering*, *2*, 373 – 388.

Chalhoub, E. S., Silva Neto, A. J., & Soeiro, F. J. C. P. (2007). Estimation of optical thickness and single scattering albedo with artificial neural networks and a monte carlo method. *Inverse Problems, Design and Optimization Symposium*, *2*, 576 – 583.

Chandrasekhar, S. (1950). *Radiative transfer*. New York: Dover Publications.

Cortivo, F. D., Chalhoub, E. S., Silva, J. D. S., & Campos Velho, H. F. (2010). Estimativa do albedo de espalhamento simples usando uma rede neural de múltiplas camadas. In *Anais...* (pp. 411 – 417). Águas de Lindóia, São Paulo: XXXIII Congresso Nacional de Matemática Aplicada e Computacional.

Dall Cortivo, F. (2008). Reconstrução de termo fonte e condições de contorno em óptica hidrológica com séries de fourier.

Dennis, J. E., & Moré, J. J. (1977). Quasi-Newton methods, motivation and theory. *Society for Industrial and Applied Mathematics Review*, *19*, 46 – 89.

Dorigo, M. (1992). Optimization, learning and natural algorithms.

Duderstadt, J. J., & Martin, W. R. (1975). *Transport Theory*. John Wiley & Sons, Inc.

Fausett, L. (1994). *Fundamentals of neural networks*. New Jersey: Prentice Hall Internetional.

Gordon, H. R. (2002). Inverse methods in hydrologic optics. *Oceanology*, *44*, 9 – 58.

Haykin, S. (1999). *Neural networks: a comprehensive foundation*. (2nd ed.). Prentice Hall.

McCormick, N. J. (1992). Inverse radiative transfer problems: a review. *Nuclear Science and Engineering*, *112*, 185 – 198.

McCormick, N. J. (2001). Inverse problems: Methods and applications. *TEMA: Tendências em Matemática Aplicada e Computacional*, *2*, 1 – 12.

McCormick, N. J. (2004). Analytic inverse radiative transfer equations for atmospheric and hydrologic optics. *Journal of the Optical Society of America.*, *21*, 1009 – 1017.

Mobley, C. D. (1994). *Light and water*. California: Academic Press.

NAG (1995). *Fortran Library Manual*. The Numerical Algorithms Group (NAG) Oxford, UK.

ISSN 2238-1007

Oliveira, R. C. (2010). Aplicação de máquinas de comitê de redes neurais artificiais na solução de um problema inverso em transferência radiativa.

Retamoso, M. R., Vilhena, M. T., Campos Velho, H. F., & Ramos, F. M. (2002). Estimation of boundary condition in hydrologic optics. *Applied Numerical Mathematics, 40*, 87 – 100.

Sansone, G. (1959). *Orthogonal functions*. New York: Interscience Publishers.

Setiono, R., & Hui, L. C. K. (1995). Use of a quasi-newton method in a feedforward neural network construction algorithm. *Neural Networks, IEEE Transactions on, 6*, 273 – 277.

Soeiro, F. J. C. P., & Silva Neto, A. J. (2006). Solution of inverse radiative transfer problems in two-layer media with artificial neural networks. In *Anais...*. Inverse Problems in Engineering Seminar.

Soeiro, F. J. C. P., Soares, P. O., & Silva Neto, A. J. (2004). Solution of inverse radiative transfer problems with artificial neural networks and hybrid method. In *Anais...*. Inverse Problems in Engineering Seminar.

Souto, R. P. (2006). Recuperação de perfis verticais de propriedade óticas inerentes a partir da radiação emergente da água. (INPE-14195-TDI/1097).

Stephany, S. (1998). Reconstrução de propriedades óticas e de fontes de bioluminescência em águas naturais.

Stephany, S., Campos Velho, H. F., & Ramos, C. D., F. M.and Mobley (2000). Identification of inherent optical properties and bioluminescence source term in a hydrologic optics problem. *Journal of Quantitative Spectroscopy & Radiative Transfer*, *67*, 113 – 123.

Vilhena, M. T., & Barichello, L. B. (1991). A new analytical approach to solve the neutron transport equation. *Kerntecnick*, *56*, 334–336.

Watrous, R. L. (1988). *Learning Algorithms for Connectionist Networks: Applied Gradient Methods of Nonlinear Optimization*. Technical Report MS-CIS-88-62 Department of Computer and Information Science, University of Pennsylvania Philadelphia, PA.

RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.