*Inverse Problems, Design and Optimization Symposium*
*Rio de Janeiro, Brazil, 2004*

# Neural Network based Models in the Inversion of Temperature Vertical Profiles from Radiation Data

**Élcio H. Shiguemori, Haroldo F. de Campos Velho**
**José Demisio S. da Silva, João C. Carvalho**

*Laboratory for Computing and Applied Mathematics – LAC*
*National Institute for Space Research – INPE*
*São José dos Campos, SP, Brazil*
[elcio, haroldo, demisio]@lac.inpe.br

## ABSTRACT

In this paper, vertical temperature profiles are inferred by a neural network based inverse procedure from satellite data. A Multilayer Perceptrons network is trained using data provided by the direct model characterized by the Radiative Transfer Equation (RTE). The neural network results are compared to the ones obtained from previous works. In addition, real radiation data from the HIRS/2 - High Resolution Infrared Radiation Sounder - is used as input for the neural networks to generate temperature profiles that are compared to measured temperature profiles from radiosonde. Analysis of the neural network results reveals the generated profiles closely approximate the results of Carvalho et al. and Ramos et al. [1, 2], thus showing adequacy of neural network based models in solving the inverse problem of temperature retrieval from satellite data. The advantages of using neural network based systems are related to their intrinsic features of parallelism and hardware implementation possibilities that may imply in very fast processing systems.

## INTRODUCTION

The vertical structure of temperature and water vapor plays an important role in the meteorological process of the atmosphere. However due to logistics and economic problems, there is a lack of observation in several regions of the Earth. In this sense, the retrieval of temperature and humidity profiles from satellite radiance data became important for applications such as weather analyses and data assimilation in numerical weather predictions models.

Interpretation of satellite radiances in terms of meteorological parameters requires the inversion of the Radiative Transfer Equation (RTE) where measurements of radiation performed in different frequencies are related to the energy from different atmospheric regions. The degree of indetermination is associated with the spectral resolution and the number of spectral channels. Moreover, usually this solution is very unstable regarding the noises in the measuring process [3, 4]. Also, several methodologies and models have been developed to improve the satellite data processing. Due to the difficulty of obtaining correct RTE solutions, several approaches and methods were developed to extract information from satellite data [5-7].

In this paper an Artificial Neural Network (ANN) is used to solve the inversion of remotely sensed data. The temperature retrievals of the new technique are compared to the ones obtained by Carvalho et al. and Ramos et al. [1, 2], who used Tikhonov and maximum entropy principle regularization techniques.

## DIRECT PROBLEM

The direct problem may be expressed by:

$$I_l(0) = B_l(T_s)\Im_l(p_s) + \int_{p_s}^{0} B_{l_i}[T(p)]\frac{\partial \Im_l(p)}{\partial p}dp, \quad (1)$$

where $I_l$ is the spectral radiance, $l$ is the channel frequency; $\Im$ is the layer to space atmospheric transmittance function, the subscript $s$ denotes surface [9]; and $B$ is the Planck function which is a function of the temperature $T$ and pressure $p$:

$$B_l(T) = \frac{2hc^2/l^5}{\left[e^{hc/k_B lT} - 1\right]} \quad (2)$$

being $h$ the Planck constant, $c$ the light speed, and $k_B$ the Boltzmann constant.

For practical purposes, equation (1) is discretized using central finite differences:

$$I_i = B_{i,s}(T_s)\Im_{i,s} + \sum_{j=1}^{N_p}\left(\frac{B_{i,j}+B_{i,j-1}}{2}\right)\left[\Im_{i,j}-\Im_{i,j-1}\right] \quad (3)$$
$$(\text{where } i=1,\ldots,N_l)$$

with $I_i \equiv I_{l_i}(0)$, $N_l$ is the number of channels in the satellite, and $N_p$ is the number of the atmospheric layers considered.

## NEURAL NETWORK ARCHITECTURE

Artificial neural networks (ANN) are made of arrangements of processing elements (*neurons*). The artificial neuron model basically consists of a linear combiner followed by an activation function. Arrangements of such units form the ANNs that are characterized by:

1. Very simple neuron-like processing elements;
2. Weighted connections between the processing elements (where knowledge is stored);
3. Highly parallel processing and distributed control;
4. Automatic learning of internal representations.

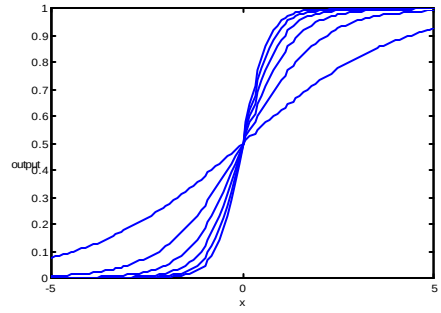
ANNs aim to explore the massively parallel network of simple elements in order to yield a result in a very short time slice and, at the same time, with insensitivity to loss and failure of some of the elements of the network. These properties make artificial neural networks appropriate for application in pattern recognition, signal processing, image processing, financing, computer vision, engineering, etc. [8-11].

The simplest ANN model is the single-layer Perceptron with a hard limiter activation function, which is appropriate for solving linear problems. This fact prevented neural networks of being massively used in the 1970s [9]. In the 1980s they reemerged due to Hopfield´s paper on recurrent networks and the publication of the two volumes on parallel distributed processing (PDP) by Rumelhart and McClelland [8].

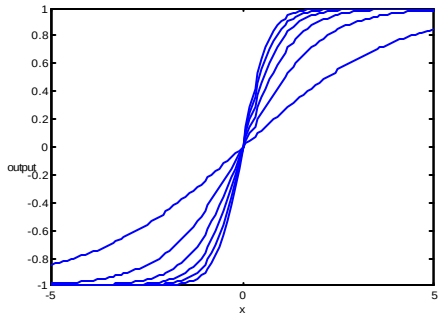
There exist different ANN architectures that are dependent upon the learning strategy adopted. This paper briefly describes the one ANN used in our simulations: the multilayer Perceptron with backpropagation learning. Detailed introduction on ANNs can be found in [8] and [11].

Multilayer perceptrons with backpropagation learning algorithm, commonly referred to as backpropagation neural networks are feedforward networks composed of an input layer, an output layer, and a number of hidden layers, whose aim is to extract high order statistics from the input data. Figure 1 depicts a backpropagation neural network with a hidden layer. Functions $g$ and $f$ provide the activation for the hidden layer and the output layer neurons, respectively. Neural networks will solve nonlinear problems, if nonlinear activation functions are used for the hidden and/or the output layers. Figure 1 shows examples of such functions.

A feedforward network can input vectors of real values onto output vector of real values. The connections among the several neurons (Figure 2) have associated weights that are adjusted during the learning process, thus changing the performance of the network. Two distinct phases can be devised while using an ANN: the training phase (learning process) and the run phase (activation of the network). The training phase consists of adjusting the weights for the best performance of the network in establishing the mapping of many input/output vector pairs. Once trained, the weights are fixed and the network can be presented to new inputs for which it calculates the corresponding outputs, based on what it has learned.




(a)




(b)

Figure 1: Two activation functions: (a) sigmoid $g(x) = \left(1+e^{-x}\right)^{-1}$; (b) $g(x) = \tanh(x) = \left(1-e^{-x}\right)\left(1+e^{-x}\right)^{-1}$.

The backpropagation training is a supervised learning algorithm that requires both input and

output (desired) data. Such pairs permit the calculation of the error of the network as the difference between the calculated output and the desired vector. The weight adjustments are conducted by backpropagating such error to the network, governed by a change rule. The weights are changed by an amount proportional to the error at that unit, times the output of the unit feeding into the weight. Equation 4 shows the general weight correction according to the so-called the delta rule

$$\Delta w_{ji} = \eta \delta_j y_i \tag{4}$$

where, $\delta_j$ is the local gradient, $y_i$ is the input signal of neuron $j$, and $\eta$ is the learning rate parameter that controls the strength of change.

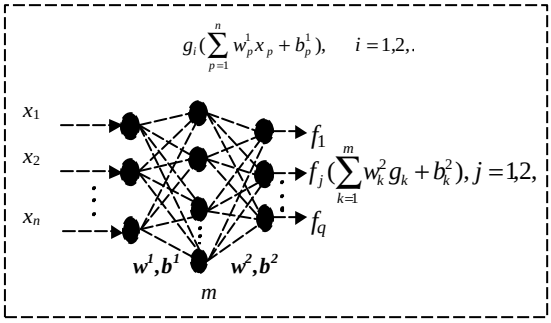



Figure2: The backpropagation neural network with one hidden layer.

## NEURAL NETWORK FOR ATMOSPHERIC PROFILE RETRIEVAL

Artificial neural networks have two stages in their application, the learning and activation steps. During the learning step, the weights and bias corresponding to each connection are adjusted to some reference examples. For activation, the output is obtained based on the weights and bias computed in the learning phase.

*Training*

The experimental data, which intrinsically contains errors in the real world, is simulated by adding a random perturbation to the exact solution of the direct problem, such that

$$\tilde{I} = I_{\text{exact}} + \sigma m \tag{5}$$

where $\sigma$ is the standard deviation of the noise and $m$ is a random variable taken from a Gaussian distribution, with zero mean and unitary variance. All numerical experiments were carried out using 5% of noise ($\sigma$=0.05).

A training set is built up from Eq. (5) and it is called SDB1 (Synthetic Dataset 1). Another dataset is used: the TIGR database – with 861 profiles, from which only 324 are chosen for the learning step. In addition, a third dataset is considered combining the both previous database (SBD1+TIGR).

Figure 1 shows the layers used for comparasion, where the error of temperature profiles is computed for each layer. This feature is important because the main interest for metheorological purposes are the layers below $p$=100 hPa.

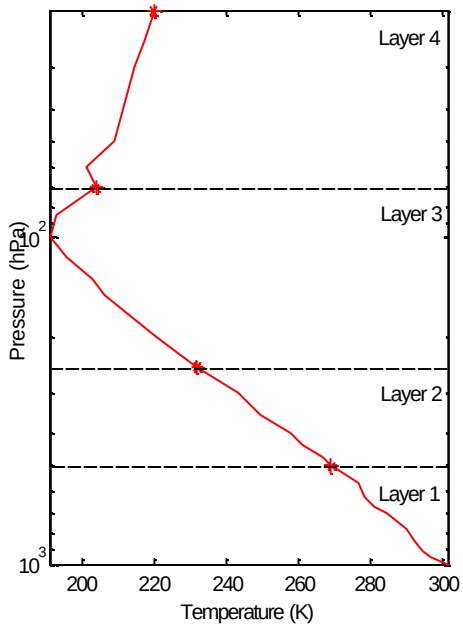



Figure 3 – Layers of atmospheric profile.

The activation test is an important procedure to indicate the permormance of an ANN, the effective test is conducted using data that does not belong to the training set. This action is called the *generalization test* of the ANN. Generalization tests are performed using 324 profiles.

The average errors of simulation results for each atmospheric layer obtained with the trained MLP are described in Tables 1, 2, and 3 – Layer-1: 20 up to 70 hPa; Layer-2: 85 up to 200 hPa, Layer-3: 250 up to 475 hPa; and Layer-4: 500 up

to 1000 hPa (see Figure 3). The error for each Layer is computed by:

$$\text{Error}=\frac{1}{N}\sum_{p_a=1}^{p_b}\sqrt{(T_i^{\text{Radiossonde}}-T_i^{\text{NeuralNetwork}})^2} \quad (4)$$

where $N$ is the number of points (sub-layers) at each layer, $p_a$ and $p_b$ are, respectively, pressure (level) at bottom and top for each layer.

Table 1: Error for activation phase using SDB1.

| ***Hidden Neurons*** | ***Layer 4*** | ***Layer 3*** | ***Layer 2*** | ***Layer 1*** |
|---|---|---|---|---|
| 1 | 4.9398 | 5.4775 | 3.5361 | 7.2428 |
| 2 | 4.0273 | 2.8786 | 2.3643 | 1.6195 |
| 3 | 3.9807 | 3.3278 | 1.8596 | 1.5637 |
| 4 | 3.9094 | 3.4567 | 2.1508 | 1.5456 |
| 5 | 3.5818 | 3.0300 | 2.2425 | 1.3647 |
| 6 | 3.2713 | 2.7158 | 2.1260 | 1.4053 |
| 7 | 3.6166 | 3.0840 | 2.0377 | 1.3938 |
| 8 | 3.3949 | 3.1996 | 1.7783 | 1.3729 |
| 9 | 3.0000 | 2.7268 | 1.9645 | 1.3984 |
| 10 | 3.3638 | 3.1020 | 1.8545 | 1.4123 |

Table 2: Error for activation phase using TIGR.

| ***Hidden Neurons*** | ***Layer 4*** | ***Layer 3*** | ***Layer 2*** | ***Layer 1*** |
|---|---|---|---|---|
| 1 | 2.4359 | 1.8599 | 2.3356 | 4.3720 |
| 2 | 3.2030 | 2.2812 | 2.6162 | 3.2077 |
| 3 | 4.1499 | 1.7741 | 2.8376 | 2.8317 |
| 4 | 5.4135 | 3.2503 | 2.7110 | 2.4988 |
| 5 | 4.6339 | 3.3412 | 4.0342 | 3.1428 |
| 6 | 5.1101 | 3.1859 | 3.8381 | 1.4613 |
| 7 | 4.5230 | 3.3218 | 4.2085 | 1.9452 |
| 8 | 4.8102 | 3.0472 | 4.2240 | 2.1983 |
| 9 | 4.7721 | 3.0709 | 4.2897 | 1.3237 |
| 10 | 4.2642 | 2.9387 | 4.5007 | 1.6613 |

Table 3: Error for activation phase using SDB1+TIGR.

| ***Hidden Neurons*** | ***Layer 4*** | ***Layer 3*** | ***Layer 2*** | ***Layer 1*** |
|---|---|---|---|---|
| 1 | 4.2138 | 4.6522 | 2.6324 | 4.8846 |
| 2 | 2.8479 | 2.5561 | 2.3623 | 1.5000 |
| 3 | 2.7113 | 2.4769 | 1.7222 | 1.5473 |
| 4 | 1.9764 | 2.5502 | 1.9703 | 1.2561 |
| 5 | 2.4640 | 2.7010 | 1.7049 | 1.2352 |
| 6 | 2.9786 | 2.5721 | 1.6046 | 1.2536 |
| 7 | 2.2562 | 2.6187 | 1.6242 | 1.1650 |
| 8 | 2.2701 | 2.5582 | 1.6117 | 1.1672 |
| 9 | 2.3689 | 2.5615 | 1.6828 | 1.3461 |
| 10 | 2.5645 | 2.5300 | 1.7036 | 1.5004 |

## SIMULATION USING REAL SATELLITE RADIANCE DATA

Simulations using real satellite radiance data, from the High Resolution Radiation Sounder (HIRS-2) of NOAA-14 satellite, have been performed to evaluate the accuracy of the Multilayer Perceptron. HIRS-2 is one of the three sounding instruments of the TIROS Operational Vertical Sounder (TOVS). ANN results are compared to *in situ* radiosonde measurements and results obtained by Carvalho et al. [1] and Ramos et al. [2], who used Tikhonov and Maximum Entropy Principle of second order regularization techniques.

The number of observations corresponds to a fraction of the number of temperatures to be estimated. For instance, in the example presented hereafter, 40 temperature values are estimated from 7 radiance measurements.

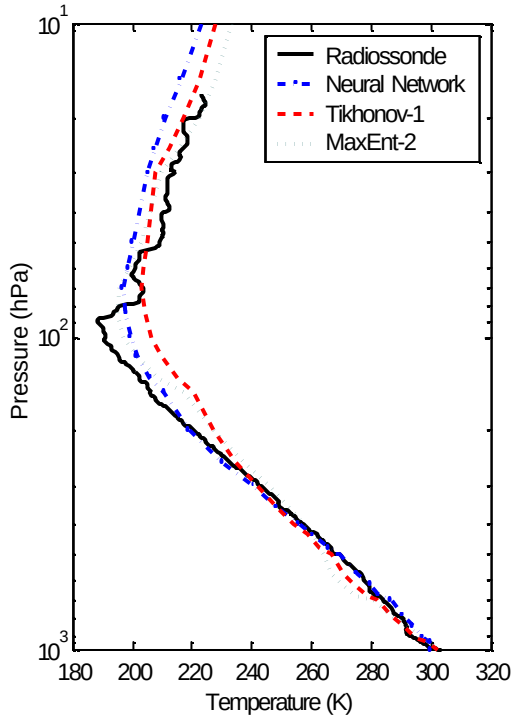



Figure 4. Retrievals achieved using radiance data from NOAA-14 satellite – ANN trained using SDB1 database.

Similar error analysis (performance of ANNs with different database for training) is also

carried out here. Tables 4, 5, and 6 present the errors associated to retrievals for ANN-MLP using different number of neurons in the hidden layer.

Table 4: Error for ANN using SDB1.

| ***Hidden Neurons*** | ***Layer 4*** | ***Layer 3*** | ***Layer 2*** | ***Layer 1*** |
|---|---|---|---|---|
| 1 | 8.3331 | 11.7801 | 4.1784 | 8.5813 |
| 2 | 4.6884 | 1.5891 | 2.0928 | 0.7541 |
| 3 | 3.9004 | 1.5897 | 2.6952 | 0.7698 |
| 4 | 4.1328 | 1.3514 | 1.9417 | 0.9395 |
| 5 | 3.4413 | 1.4532 | 1.6697 | 0.7453 |
| 6 | 3.4493 | 2.0544 | 0.4835 | 1.1023 |
| 7 | 3.8532 | 1.2495 | 1.2234 | 0.5779 |
| 8 | 3.1445 | 1.6497 | 2.1032 | 0.7418 |
| 9 | 2.7738 | 2.1136 | 1.6062 | 0.6293 |
| 10 | 3.2128 | 2.1856 | 0.6496 | 0.6350 |

Table 5: Error for ANN using TIGR.

| ***Hidden Neurons*** | ***Layer 4*** | ***Layer 3*** | ***Layer 2*** | ***Layer 1*** |
|---|---|---|---|---|
| 1 | 3.6970 | 6.4571 | 4.7807 | 3.9290 |
| 2 | 1.0035 | 3.4671 | 1.8482 | 1.6938 |
| 3 | 1.3682 | 4.2400 | 1.6443 | 1.5805 |
| 4 | 1.2431 | 3.8797 | 1.6800 | 1.4246 |
| 5 | 1.2396 | 3.4584 | 1.9324 | 1.2227 |
| 6 | 1.2302 | 3.4992 | 2.1354 | 1.4285 |
| 7 | 1.2052 | 3.5170 | 2.1433 | 1.2738 |
| 8 | 1.5869 | 4.0933 | 2.2418 | 1.3683 |
| 9 | 1.2290 | 3.5378 | 2.2657 | 1.3823 |
| 10 | 1.2313 | 3.6360 | 2.3371 | 1.4322 |

Table 6: Error for ANN using SDB1+TIGR.

| ***Hidden Neurons*** | ***Layer 4*** | ***Layer 3*** | ***Layer 2*** | ***Layer 1*** |
|---|---|---|---|---|
| 1 | 7.2495 | 10.5682 | 2.9538 | 5.9283 |
| 2 | 2.7197 | 3.1742 | 3.5973 | 1.0311 |
| 3 | 2.5389 | 3.6098 | 1.7893 | 0.8741 |
| 4 | 1.4912 | 4.9095 | 2.0079 | 0.6776 |
| 5 | 1.8251 | 4.1844 | 0.8758 | 0.5170 |
| 6 | 2.9004 | 2.9490 | 1.5373 | 0.8462 |
| 7 | 1.3932 | 4.1497 | 2.1490 | 0.6573 |
| 8 | 1.6752 | 3.7681 | 2.6354 | 0.7627 |
| 9 | 1.9158 | 4.0173 | 1.5537 | 0.8827 |
| 10 | 2.2663 | 3.7773 | 1.2204 | 1.3050 |

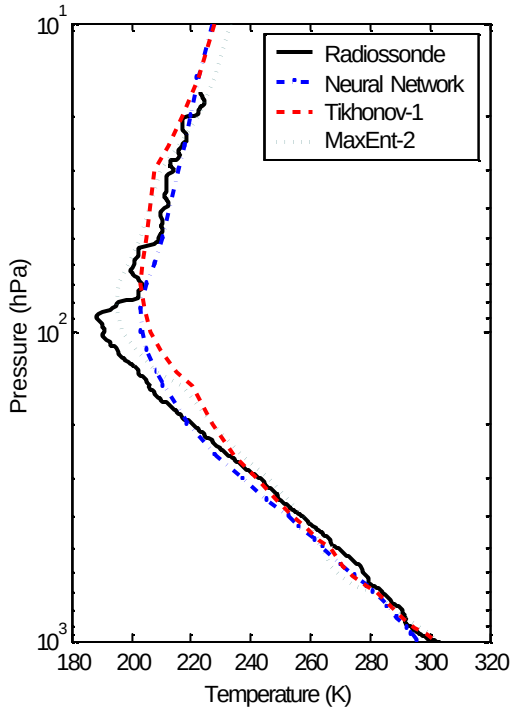



Figure 5. Retrievals achieved using radiance from NOAA-14 satellite – result for TIGR database.

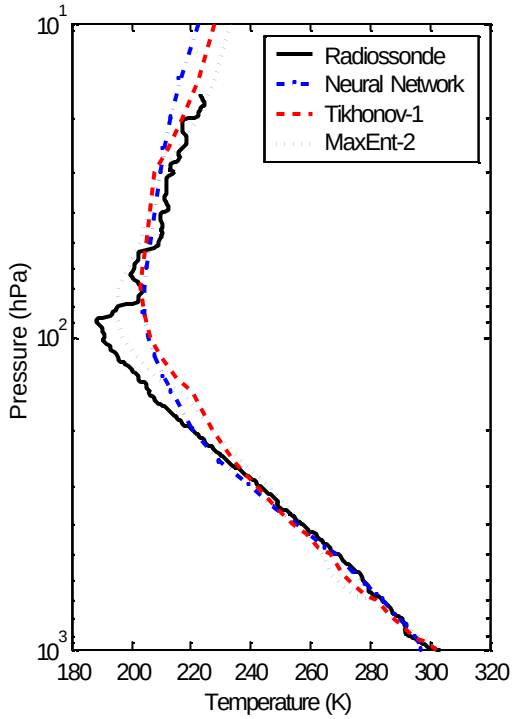



Figure 6. Retrievals achieved using radiance data from NOAA-14 satellite – SDB1+TIGR dataset.

## CONCLUSION

The mathematical formulation of the problem of retrieving vertical temperature profiles from remote sensing data is given by the integral radiative transfer equation, and leads to the solution of a highly ill-conditioned Fredholm integral equation of the first kind. Temperature versus atmospheric pressure plots are presented in Figures 4, 5 and 6. The results show the good agreement between retrievals from the ANNs and the radiossonde measurements. It should be noted that the ANN trained with SBD1 dataset presented the best performance, in general, because the temperature profiles used during the learning phase are closer to the regional climate mean. However, combining the SDB1 with the TIGR dataset improved the retrievals. This is an evidence of the database influence for the multi-layer perceptron neural network for the atmospheric temperature inversion.

ANNs were effective for solving this inverse problem, and the reconstructions are comparable with those obtained with regularization methods [1, 2], even for data containing noise. However, the NN do not remove the inherent ill-posedness of the inverse problem.

In practice, operational inversion algorithms reduce the risk of being trapped in local minima by starting the iterative search process from an initial guess solution that is sufficiently close to the true profile. However, the dependence of the final solution on a good choice of the initial guess represents a fundamental weakness of such algorithms, particularly in regions where less a priori information is available [12]. ANN can relax this constrain incorporating more data in the dataset during the learning phase.

Some advantages can be listed with the use of ANNs: after the training phase, the inversion with NNs is much faster than the regularization methods; it is an intrinsicly parallel algorithm; finally, ANNs can be implemented in hardware devices, the neurocomputers, becoming the inversion processing faster than ANNs emulated by software.

Other architectures for ANNs deserve to be investigated. One preliminar result can be seen with non-linear Hopfield's neural network [13].

## REFERENCES

1. J.C. Carvalho, F.M. Ramos, N.J. Ferreira H.F. de Campos Velho, Retrieval of Vertical Temperature Profiles in the Atmosphere, *3rd International Conference on Inverse Problems in Engineering* (3ICIPE), Proceedings in CD-ROM, under paper code HT02 (1999) – Proc. Book: pp. 235-238, Port Ludlow, Washington, USA, June 13-18, UEF-ASME (2000).

2. F.M. Ramos, H.F. de Campos Velho, J.C. Carvalho, N.J. Ferreira, Novel Approaches on Entropic Regularization, *Inverse Problems*, **15**(5), 1139-1148,(1999).

3. C. D. Rodgers, Retrieval of the atmospheric temperature and composition from remote measurements of thermal radiation, *Rev. Geophys. Space Phys.*, **14**, 609-624 (1976).

4. S. Twomey, *Introduction to the matehmatics of inversion in remote sensing and interative measurements*, Amsterdam, Elsevier Scientific, 1977.

5. M. T. Chahine, Inverse Problem in Radiative Transfer: determination of atmospheric parameters, *Jour. Atmos. Sci*., **27**, 960 (1970).

6. W. L. Smith, H. M. Woolf, A. J. Schriener, Simultaneous retrieval os surface and atmospheric parameters: a physical analytically direct approach, *Adv. In Rem. Sens.*, **7** (1985).

7. K. N. Liou, *An introduction to atmospheric radiation, academic press*, Orlando, 1982.

8. S. Haykin, *Neural Networks: A Comprehensive Foundation*, Macmillan. New York, 1994.

9. C-T Lin and G. Lee, *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*, Prentice Hall, New Jersey, 1996.

10. M. Nadler and E.P. Smith, *Pattern Recognition Engineering*, John Wiley & Sons, New York, 1993.

11. L.H. Tsoukalas and R.E. Uhrig, *Fuzzy and Neural Approaches in Engineering*, John Wiley & Sons, New York, 1997.

12. A. Chédin, N. A. Scott, C. Wahiche, P. Moulini, The improved initialization inversion method: A high resolution physical method for temperature retrievals from TIROS-N series. *J. Climate Appl. Meteor*, **24**, 128-143.

13. V.C. de Viterbo, J.P. Braga, E.H. Shiguemori, J.D.S. da Silva, H.F. de Campos Velho, Atmospheric temperature retrieval using non-linear Hopfield neural network, *Inverse Problems, Design and Optimization Symposium* (2004) – in this Proceedings.