Automatic configuration for neural network applied to atmospheric temperature profile identification

Sabrina B. M. Sambatti
National Institute for Space Research, Brazil

Juliana Anochi
National Institute for Space Research, Brazil

Eduardo Fávero Pacheco da Luz
Centro Nacional de Monitoramento e Alertas de Desastres Naturais

Adenilson Carvalho
Departamento de Ciência e Tecnologia Aeroespacial

EngOpt 2012 - International Conference on Engineering Optimization
Rio de Janeiro, Brazil, 1-5 July 2012.

Automatic configuration for neural network applied to atmospheric temperature profile identification

Sabrina Bergoch Monteiro Sambatti[1], Juliana Aparecida Anochi[1], Eduardo Fávero Pacheco Luz[1], Elcio Hideiti Shiguemori[3], Adenilson R Carvalho[2], Haroldo Fraga de Campos Velho[1]

[1] National Institute of Space Research - INPE, São José dos Campos, SP, Brazil
[2] Data Processing Federal Service - SERPRO, São Paulo, SP, Brazil
[3] Institute for Advanced Studies - IEAv, São José dos Campos, SP, Brazil

Abstract

Multi-particle collision algorithm (MPCA) is applied to design an optimum architecture for a supervised ANN. The MPCA optimization algorithm emulates a collision process of multiple particles inspired in processes of a neutron traveling in a nuclear reactor. The procedure to carry out the automatic configuration for multi-layer perceptron (MLP) neural network is applied to identify the vertical temperature profiles are obtained from measured satellite radiance data. The MLP-NN is trained with data provided by the direct model characterized by the Radiative Transfer Equation (RTE). The MLP-NN results are compared to the ones computed using regularized inverse solutions. In addition to synthetic data (corrupted by noise), real radiation data from the HIRS/2 (High Resolution Infrared Radiation Sounder) is used as input for the MLP-NN to generate temperature profiles that are compared with the temperature profiles measured by a radiosonde. The comparison between the results obtined with automatic process and previous configuration chosen by an expert is evaluated.

Keywords: Atmospheric temperature profile, artificial neural network, optimized topology, inverse problem.

1 Introduction

During the last decades, artificial neural networks (ANN) models have been one of the most techniques commonly used from the Artificial Intelligence, and nowadays, it is under intense research worldwide. Although there are a lot research in this area, there are still many questions about the ANN models that need to be better addressed. One of the relevant research topic on ANN models is the optimal or close optimal architecture. The process of obtaining an appropriated neural network topology to solve a specific problem is a complex task, and usually requires a great effort from the designer to determine the best parameters, and it is also necessary a previous knowledge about the problem to be treated.

The process for searching and definition of an optimal architecture for an ANN is very relevant, demanding an intensive research about computational efficiency of the model [1].

There are many algorithms in the literature for the ANN training aimed the improving of the ability of generalization and for the control of an adequate architecture specification, such as: *Pruning* [3]: makes adjustment of neural network by modifying its structure, the training begin with an oversized architecture and the weights are eliminated until the capacity of generalization can be increased; *Weight Decay* suggested by [3]: the algorithm is similar to the *Pruning*, where the cost function and weight vector are modified; *Early Stopping* proposed by [11]: the scheme performs the early interruption of training, without changing the ANN architecture; *Cross Validation* proposed by [9]: it is known to improve the generalization, where the data set is separated in two data sets, training data and validation data, the validation set is responsible for evaluating the performance of the models.

The training algorithms based on techniques of multi-objective optimization have been applied for the learning. This strategy is an alternative to address the problem of finding an optimal architecture and to present a good capacity of generalization. Reference [10] presents a new learning algorithm to improve the generalization of the model of Multi-Layer Perceptron (MLP). This algorithm uses the training techniques of multi-objective optimization, which proposes to control the complexity of neural networks using simultaneous minimization of training error and norm of weight vector. The authors argue the necessity of the use of more than one objective function for dealing with the problem of supervised learning. Braga [12] presents a new constructive method and pruning approaches to control the design of Multi-Layer Perceptron without loss in performance. The proposed methods use a multi-objective approach to guarantee generalization. The pruning methods are able to simplify the network topology and to identify linear connections between the inputs and outputs of the neural model.

The computational complexity of the architecture of a neural network can be based on the number of neurons and the number of epochs, as proposed by [1]. This feature is applied to design an ANN, and it is employed to the problem of recovering atmospheric profiles of gas concentration (or the soil source term for emission or absorption). The objective function is the summation of square difference (between the target and the ANN output) and complexity.

This paper proposes a method based on stochastic optimization techniques to identify optimal architecture for an ANN. A penalty term is used to evaluate the objective function to avoid a very complex network architectures. The minimization of this function involves the balance between the training error and generalization error. Multiple Particle Collision Algorithm (MPCA) is the optimization algorithm used.

The MPCA is a meta-heuristic proposed by Luz [4], and here it is applied for optimization of ANN models. The parameters to be estimated in the worked ANN are: number of hidden layers, number of artificial neurons, type of activation function, learning and momentum rates.

An Artificial Neural Network (ANN) is used to solve the inversion using satellite data, designed by employing the MPCA. The temperature retrievals obtained with automatic configuration are compared to the ones obtained by Shiguemori et al. [12], where an empirical ANN was used to configure the ANN.

2 Artificial Neural Network

Artificial Neural Networks are computational techniques that present a mathematical model inspired by the neural structure of biological organisms, acquiring knowledge from the experience.

ANN are parallel distributed systems, composed of neurons or processing units, which calculates certain mathematical functions, typically nonlinear. These neurons can be divided into one or more layers interconnected by synaptic weights (connections), which store the knowledge represented in the model and serve to balance the input received by each network neuron [2].

The artificial neuron model basically consists of a linear combiner followed by an activation function given by:

$$y_k = \varphi\left(\sum_{j=1}^{n} w_{kj}\, x_j + b_k\right) \tag{1}$$

where w_{kj} are the connections weights, b_k is a threshold parameter, x_j is the input vector and y_k is the output of the k^{th} neuron, the $\varphi(\cdot)$ is the function that provides the activation for the neuron. Neural networks will solve nonlinear problems, if nonlinear activation functions are used for the hidden and/or the output layers.

The property of primary significance for a neural network is the ability to learn from the environment, improving its performance through the learning. A neural network learns from its environment through an interactive process of adjustments applied to its synaptic weights and bias levels [2]. In the context of neural networks, the training process can be defined as a set of well defined rules for solving a specific problem of learning.

The training algorithms are divided into two classes: supervised learning and unsupervised learning. In the supervised learning, input pattern and the desired output are provided by an external supervisor to set the parameters of the network, in order to find a connection weight between pairs of input and provided output. For this type of training, the output is compared with the desired response and the weights of connections are adjusted by minimizing the error. In the training with unsupervised learning, only the input patterns are presented to the network: there is no an external supervisor to indicate the desired output for input patterns.

2.1 Multilayer Perceptron Network

Multilayer perceptrons have been applied successfully to solve several and difficult problems by training them with a supervised manner with a highly popular backpropagation algorithm. This algorithm is based on the error correction learning rule.

The backpropagation learning error consists of two steps through the different layers of the network: a forward step and a backward step. For the forward step, an activity pattern (input vector) is applied thought the nodes of the network, and its effect propagates on the entire the network, layer by layer. Finally, a set of outputs is produced as the actual responde of the network. During the backward step, the synaptic weights are all adjusted in accordance with an error correction rule. The response of the network is subtracted from a desired responde to produce an error signal. This way, the error signal is backward neuron through the network, against the direction of synaptic connections. The scheme is named the backpropagation error [2].

Figure 1 shows the architecture of a MLP network comprising: an input layer, where the patterns are presented to the network, an intermediate layer, which works as a recognizer of characteristics that are storaged in the synaptic weights and account for most of the processing, and an output layer, where the results are presented.

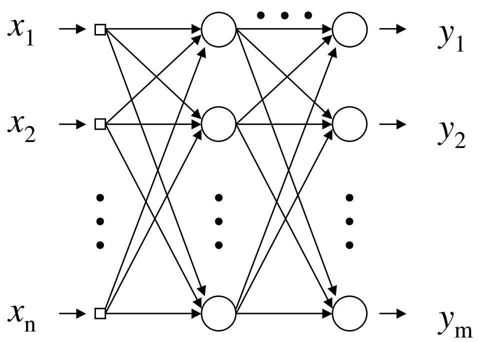

Figure 1: Multilayer Neural Network

In order to evaluate the performance of the ANN models, the mean square error is used:

$$E_{gen} = \frac{1}{N} \sum_{k=1}^{N} (y_k - \hat{y_k})^2 \tag{2}$$

where N is the number of grid points, y_k is the true observational value, and $\hat{y}_k$ is the estimation computed by the neural model.

3 Multiple Particle Collision Algorithm

High-performance computing systems are a reality in several research centers. Some of these systems are able to provide up to petaflops, enabling the solution of large problems through new algorithms that can take advantage of these environments.

The Multi Particle Collision Algorithm (MPCA) is a stochastic optimization method developed by [4], and the algorithm was codified to run in a high-performance computing environments. This version of the standard PCA [8] uses multiple particles in a collaborative way, organizing a population of candidate solutions. The PCA was inspired by the traveling process (with absorption and scattering) of a particle (neutron) in a nuclear reactor. The use of the PCA was effective for several test functions and real applications [7].

The PCA starts with a selection of an initial solution (Old-Config), it is modified by a stochastic perturbation (*Perturbation*{.}), leading to the construction of a new solution (New-Config). The new solution is compared (function *Fitness*{.}), and the new solution can or cannot be accepted. If the new solution is not accepted, the scheme of scattering (*Scaterring*{.}) is used. The exploration around closer positions is guaranteed by using the functions *Perturbation*{.} and *Small-Perturbation*{.}. If the new solution is better than the previous one, this new solution is absorbed. If a worse solution is found, the particle can be sent to a different location of the search space, it enables the algorithm of escaping a local minimum [5].

The implementation of the MPCA algorithm is similar to PCA, but it uses a set with N particles, where a mechanism to share the particle information is necessary. A blackboard strategy is adopted, where the best-fitness information is shared among all particles in the process. This process was implemented in Message Passing Interface (MPI), looking for application into a distributed memory machine [6]. The pseudo-code for the MPCA is presented by Table 1.

Table 1: MPCA: pseudo-code for the algorithm.

```
Generate an initial solution: Old-Config
Best-Fitness = Fitness{Old-Config}
Update Blackboard
For n = 0 to # of particles
   For n = 0 to # iterations
   Update Blackboard
   Perturbation{.}
      If Fitness{New-Config} > Fitness{Old-Config}
         If Fitness{New-Config} > Best-Fitness
            Best-Fitness = Fitness{New-Config}
         End If
         Old-Config = New-Config
         Exploration{.}
      Else
         Scattering{.}
      End If
   End For
End For
```

4 Configuring the MLP-NN by MPCA

An ANN architecture is not previously known. Usually, the best architecture is empirically determined. However, the problem of identification of an optimized ANN architecture can be formulated as a search in the space of solutions, where each point represents a possible architecture. If a performance value is associated with each point or solution, in such a way that this value is based on some optimality criterion (complexity), it is possible to construct a hyper-surface, where the highest point (or the lowest) is equivalent to the best architecture. Therefore, the problem can be treated as an optimization problem, and the goal is to find the optimum value in this surface, which represents the best combination of variables [1].

This paper uses an stochastic method called MPCA. The method is enable to make the balance of the behavior between global search (exploration) and local search (exploitation), this balance is essential to prevent that the search will not be stop in a local optimum, enabling the searching for global optimum [4].

The optimization problem is formulated by an objective function and a set of restrictions that need to be satisfied. The objective function used in this article is a combination of two factors: square difference between the target values and the ANN output, and a penalty factor. The latter factor is expressed by [1]:

$$f_{\text{obj}} = penalty \times \left(\frac{\rho_1 \times E_{train} + \rho_2 \times E_{gen}}{\rho_1 + \rho_2} \right) \tag{3}$$

where $\rho_1 = 1$ and $\rho_2 = 0.1$ are factors that modify the relevance allowed to the training and generalization error. There is great flexibility in the evaluation of the objective function, because the training error is directly related to the network memory capacity and generalization error refers to the ability of the ANN to identify the patterns that are similar to ones used in training. The function f_{obj} consists of the sum of squared errors for training and generalization multiplied by the penalty, who is responsible by the complexity of neural network architecture in question. The minimum value of f_{obj} corresponds to a simple architecture that displays consistent behavior in the solution space combined with low training error and generalization. Thus, it is a simple architecture, where the total weights and bias time of learning can be reduced [**?**].

The MPCA is employed to evolve: (i) the number of neurons in the intermediate (hidden) layer, (ii) the learning rate parameter η, (iii) momentum constant α. Allowed values for these parameters are shown in Table 2

Table 2: Parameters to define a network architecture.

Parameter	Value
Neuron in the hidden layer	| 1 | ... | 32 |
Learning ratio: η	| 0.0 | ... | 1.0 |
Momentum constant: α	| 0.1 | ... | 0.9 |

The MPCA is used to generate a set of candidate solutions that correspond to an ANN architecture. For each solution, the ANN is activated, and the training process starts until the stopping criterion is satisfied (error minimum or total epochs). With the values obtained by ANN, the MPCA calculates the objective function, up dating the parameters for the ANN. This process is repeated until an optimal value for the objective function is found.

5 Neural Network for Atmospheric Profile Retrieval

Regularized inverse solution deals with the problem of retrieving vertical temperature profiles from remote sensing data, where the integral radiative transfer equation leads to the inverse solution of a highly ill-conditioned Fredholm integral equation of the first kind. For this paper, the inverse solution is computed by using a MLP-NN.

5.1 Data for training

The experimental data is simulated by adding a random perturbation to the exact solution of the direct problem, given by:

$$\tilde{I} = I_{\text{exact}} + I_{\text{exact}} \sigma \mu \tag{4}$$

where σ is the standard deviation of the noise and μ is a random variable taken from a Gaussian distribution, with zero mean and unitary variance. All numerical experiments were carried out using σ=0.05.

The direct problem is described by the radiative transfer equation (RTE). It is described using the linear integro-differential Boltzamnn equation [13]. However, depending on the range of satellite observation – infrared, in our case – the RTE can be simplified. The Schwarzchild's equation is a RTE version where the scattering phenomenon can be neglected, having a local thermodynamic equilibrium. This means that the atmosphere behaves as a black body, following the Plank's law, relating the radiances with the body temperature. The calculation of radiance values from associated temperatures is expressed by [13]:

$$I_\lambda(0) = B_\lambda(T_s)\Im_\lambda(p_s) + \int_{p_s}^{0} B_\lambda[T(p)]\frac{\partial \Im_\lambda(p)}{\partial p}dp \tag{5}$$

where, I_λ is the value of the spectral radiance, subscript s denotes surface; p is the considered pressure; $\Im$ is the transmittance for a given atmospheric layer that is a function of the wavelength and the concentration of absorbent gas, which usually decreases exponentially with the height. In pressure coordinate, the transmittance may be expressed by:

$$\Im_\lambda(p) = \exp\left[-\frac{1}{g}\int_{p_0}^{p} k_\nu(p)q(p)dp\right] \tag{6}$$

where k_ν is the absorption coefficient, q is the ratio of gas mixture, g is the acceleration of the local gravity, and p_0 is the pressure on the top of the atmosphere; B is the Planck's function (Eq. 7), which is a function of the temperature T and wavelength λ:

$$B_\lambda(T) = \frac{2hc^2/\lambda^5}{[e^{hc/k_B\lambda T} - 1]} \tag{7}$$

with h being the Planck's constant; c is the light speed; and k_B is the Boltzmann's constant. Equation (5) is discretized using central finite differences:

$$I_i = B_{i,s}(T_s)\Im_{i,s} + \sum_{j=1}^{N_p}\left(\frac{B_{i,j} + B_{i,j-1}}{2}\right)[\Im_{i,j} - \Im_{i,j-1}] \tag{8}$$

where $i = 1, ..., N_\lambda$, $I_i \equiv I_{\lambda_i}$, N_λ is the number of satellite channels, $B_{i,j} = B_{\lambda_i}(T_j)$, $\Im_{i,j} = \Im_{\lambda_i}(p_j)$, and N_p is the number of atmospheric layers considered. It is assumed that each atmospheric layer has a characteristic (constant) temperature T_j to be computed.

A training set is built up from Eq. (4) 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 2 [12] shows the layers used for comparasion, where the error of temperature profiles is computed for each layer, with Layer-1 [1000 hPa, 500 hpa]; Layer-2 [500 hPa, 250 hpa]; Layer-3 [250 hPa, 85 hpa]; Layer-4 [85 hPa, 20 hpa]. This feature is important because the main i nterest for metheorological purposes are the layers below $p = 100$ hPa, where 1 hPa = 100 Pa.

6 Estimation Using Real Satellite Radiance Data

Inversion dealing real satellite radiance data, from the High Resolution Radiation Sounder (HIRS-2) of the NOAA-14 satellite, have been performed to evaluate the accuracy of the multilayer perceptron neural network. HIRS-2 is one of the three sounding instruments from the TIROS-TOVS (Operational Vertical Sounder) satellite system. Results for ANN designed by MPCA are compared to *in situ* radiosonde measurements and results obtained by Shiguemori [12], who obtained an empirical ANN configuration.

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.

The results presented in this section take in consideration the average of 10 experiments with seeds generate different random numbers and experimental data generated artificially. The parameters used are: 4 particles; 4 processors; 500 iterations. The stopping criterion used was the maximum number of evaluations of the objective function. The algorithms are implemented in Fortran 90, and computer tests were conducted under Linux operating system. The best network architecture found corresponds to the network parameters configuration shown in the Table 3.

Figure 3 shows the results obtained with the two architecture configurations determined by MPCA and by empirical configuration (performed by an expert). As it can be seen, the results are in a good

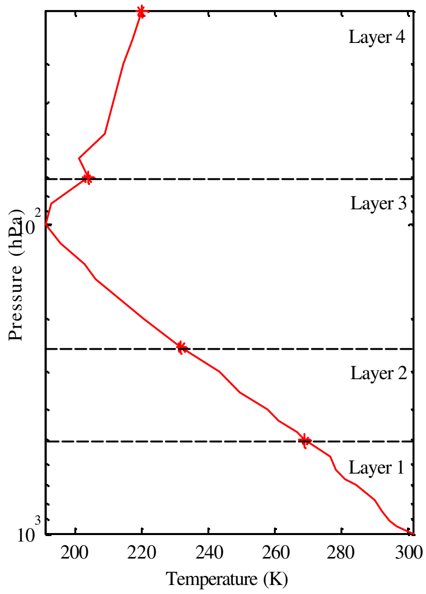

Figure 2: Layers of atmospheric profile

Table 3: Configuration Architecture

Parameter	ANN MLP - MPCA	ANN - Empirical
Hidden Layers	1	1
Neuron in 1 hidden layer	1	7
Learning rate η	0.0054	0.4
Constant momentum α	0.136	0.6
Activation function	*Logistic*	*Logistic*
Mean Square Error	0.000014	0.5170

agreement. The procedure of an automatic method to identify a good configuration for the ANN does not require the supervision of an expert. It is possible to notice a reasonable agreement between ANN's retrievals and the radiosonde measurements.

7 Conclusions

This paper deals with an automatic scheme to identify the ANN architecture. The problem is formulated as an optimization process. The stochastic technique MPCA was employed to address the solution of the optimization problem.

The scheme to identify an automatic ANN configuration was applied to atmospheric temperature profile identification. The results are shown in figure 3, where the good performance to determine an appropriate ANN is verified.

The automatic method to identify a configuration for the ANN does not require the help of an expert. This special issue allows the application of ANN technique for a larger community. Additionally, some solution can be found that could never be tested by a human being.

Acknowledgements

The authors thank to the CNPq, CAPES, and FAPESP for the financial support.

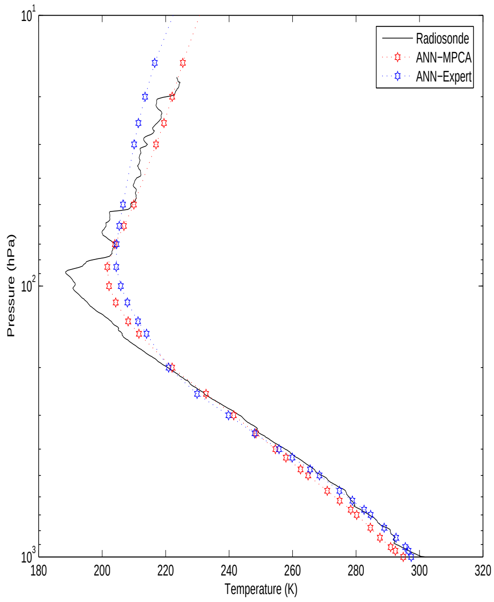

Figure 3: Retrievals achieved using radiance data from NOAA-14 satellite - ANN:MPCA trained using SDB1+TIGR database

8 References

[1] Carvalho A. R., "Uso de redes neurais otimizadas para recuperação do perfil de concentração de gases traços atmosféricos a partir de dados de satélites". *D.Sc. thesis on Applied Computing – Instituto Nacional de Pesquisas Espaciais*, June-2011.

[2] Haykin S., "Neural Networks: A Comprehensive Foundation". *Prentice-Hall Inc.*, 842p, 1999.

[3] Hinton G. E., "Connectionist Learning Procedures". *Artificial Intelligence*, 185-234, 1989.

[4] Luz E. F. P., Becceneri J.C., Campos Velho H.F., "A new multi-particle collision algorithm for otimization in a high-performance environment". *Journal of Computacional Interdisciplinary Sciences*, **1**, 1-7, 2008.

[5] Luz E. F. P., "Meta-heurísticas paralelas na solução de problemas inversos". *D.Sc. thesis on Applied Computing – Instituto Nacional de Pesquisas Espaciais*, March-2012.

[6] Sambatti S. B. M., "Diferentes estratégias de paralelização de um algoritmo genético epidêmico aplicadas na solução de problemas inversos". *M.Sc. dissertation on Applied Computing – Instituto Nacional de Pesquisas Espaciais*, 2004.

[7] Sacco W. F., Knupp D. C., Luz E. F. P. da., Silva Neto A. J., "Algoritmo de Colisão de Partículas". *Técnicas de Inteligência Computacional Inspiradas na Natureza Aplicação em Problemas Inversos em Transferência Radiativa*, 79-90, 2009.

[8] Sacco W. F., Oliveira C. R. E. A., "A new stochastic optimization algorithm based on a particle collision metaheuristic.". *6th World Congress of Structural and Multidisciplinary Optimization –* WCSMO, Rio de Janeiro, 2005.

[9] Stone M., "Cross-validation A review". *Mathematische Operationforsch Statischen*, 127-139, 1978.

[10] Texeira R. A., Braga A. P., Takahashi R. H. C., Saldanha R. R., "Improving generalization of MLP with multi-objective optimization". *Neurocomputing*, 189-19, 2000.

[**11**] Weigend A. S., Huberman B., "Predicting the Future: a Connectionist Approach". *International Journal of Neural Systems*, 193-209, 1990.

[**12**] Shiguemori E. H., Velho H. F. D. C., Silva J. D., Carvalho J. C., "Neural network based models in the inversion of temperature vertical profiles from radiation data". *Inverse Problems in Science and Engineering*, 1-16, 2005.

[**13**] Liou K. N., "An introduction to atmospheric radiation". *Academic Press*, Orlando, 1982.

[**14**] Costa M. A., Braga A.P., Menezez B. R., "Improving neural networks generalization with new constructive and pruning methods". *Journal of Intelligent and Fuzzy Systems*, 75?83, 2003.