MPCA META-HEURISTICS FOR AUTOMATIC ARCHITECTURE OPTIMIZATION OF A SUPERVISED ARTIFICIAL NEURAL NETWORK

S. B. M. Sambatti[1], J. A. Anochi[1], E. F. Pacheco da Luz[1], A. R. Carvalho[2], E. Shiguemori[3], H. F. Campos Velho[1]

[1] National Institute of Space Research - INPE
[2] Data Processing Federal Service - SERPRO
[3] Institute for Advanced Studies - IEAv

Abstract. *Artificial neural networks (ANN) has been studied intensively, but there still are many unresolved issues. The search and definition of an optimal architecture remains a very relevant ANN research topic. The search space of neural network topology, each point represents a possible architecture. Associating each point to a performance level relies on the a priori establishment of some optimality criterion. Here, a new meta-heuristics, multi-particle collision algorithm (MPCA) was 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 multilayer perceptron (MLP) was the neural network adopted here, and backpropagation strategy was used for calculating of the weight of connections to the MLP-NN. The MLP-NN configured by this optimal or inverse designs was applied to predict the seasonal mesoscale climate. The dataset for trainning is obtained from NCEP-NOAA reanalysis and from a metherological model. In order to reduce the dimension of the search space to find the optimized ANN, it is considered the following: three activation functions, up to three hidden layers, and up to 32 neurons per hidden layer. The comparison is performed between the ANN configuration obtained by automatic process and another configuration proposed by a human specialist.*

1. INTRODUCTION

During the last decades, artificial neural networks (ANN) models have been one of the most tecniques 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 treated. One of the main topics on ANN models is the architecture complexity optimal or close optimal for the training process. The process of obtaining an adequate neural network to solve a specific problem is a complex task and usually requires a great effort by the designer mainly to determine the best parameters, and it is 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 [2].

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* [4], 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 [4], the algorithm is similar to the *Pruning*, where the cost function and weight vector are modified; *Early Stopping* proposed by [13], the scheme performs the early interruption of training, without changing the ANN architecture, similarly; *Cross Validation* proposed by [10], it is known to improve the generalization, where the data set is separated in two 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 [11] presents a new learning algorithm to improve the generalization of the model of Multiple Layers 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 need of the use of more than one objective function for dealing with the problem of supervised learning.

The computational complexity of the architecture of a neural network based on the number of neurons and the number of epochs is proposed by [2]. This feature is applied to design an ANN, and it is employed to the problem of recovering atmospheric profiles of gas concentration. The objective function to be solved by optimization techniques 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 in the objective function to avoid 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 [5], and here it is firstly 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 rate and momentum.

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 through 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 [3].

The artificial neuron model proposed by McCullock and Pitts in 1943, interprets the functioning of the neuron as a simple binary circuit that combines multiple inputs and one output signal. The mathematical description resulted in a model with n input terminals representing the dendrites, and only one output simulating the axon.

In the mathematical terms, a neuron k could be described considering the following pair of equations [3]:

$$v_k = \sum_{j=1}^{n} w_{kj}\, x_j \quad (1)$$

$$y_k = \varphi(v_k + b_k) \quad (2)$$

where $x_1, x_2, ..., x_n$ are the input signals; w_{kj} are the synaptic weights of neuron k; v_k is the linear combination among the input signals; b_k is the bias; φ is the activation function; and y_k is the output signal of the neuron. The use of bias b_k has the effect of applying an affine transformation to the output v_k of the linear combiner in the model.

There are many activation functions that can be applied to create distinct neurons. The following functions are used here: Gaussian function, logistic and hyperbolic tangent function, they are shown in Figure 1:

Figure 1. Graphical representation of different activation functions: (a) Gaussian function; (b) logistic function; (c) hyperbolic tangent.

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 [3]. In the context of neural networks, the learning 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 output provided. 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 some difficult and diverse problems by training them with a supervised manner with a highly popular backpropagation algorithm. This a;gorithm 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 [3].

Figure 2 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 2. Multilayer Perceptron Network

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

$$E_{gen} = \frac{1}{N} \sum_{k=1}^{N} (y_k - \hat{y_k})^2 \tag{3}$$

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 [5], and the algorithm was preparated to run in a high-performance computing environments. This version of the standard PCA [9] uses multiple particles in a colaborative way, organizing a population of candidate solutions. The PCA was inspired by the scattering of a particle in a nuclear reactor. The use of the PCA was effective for several test functions and real applications [8].

The PCA starts with a selection of an initial solution (Old-Config), it is modified by a sthocastic perturbation (*Perturbation*$\{.\}$), leading to the construction of a new solution (New-Config). The new solution is compared (function *Fitness*$\{.\}$), and the new solution can ou cannot be accepted. If the new solution is not accepted, the scheme of scaterring (*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 send to a different location of the search space, it enables the algorithm of escaping a local minimum [6].

The implementation of the MPCA algorithm is similar to PCA, but it uses a set with *n* particles, where a mechanism to share the particles 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 [7]. The pseudo-code for the MPCA is presented by Table 1.

Table 1. MPCA: psedo-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. METHODOLOGY

4.1. 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 criterium (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 [2].

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 essencial to prevent that the search will not be stop in a local optimum, enabling the searching for global optimum [5].

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 [2]:

$$f_{obj} = penalty \times \left(\frac{\rho_1 \times E_{train} + \rho_2 \times E_{gen}}{\rho_1 + \rho_2} \right) \tag{4}$$

where ρ_1 e ρ_2 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 ANN to identify the patterns that are similar to patterns used in training.

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

Table 2. Parameters to define a network architecture.

Parameter	Value
Hidden Layers	$\mid 1 \mid 2 \mid 3 \mid$
Neuron in each hidden layer	$\mid 1 \mid \ldots \mid 32 \mid$
Learning ratio: η	$\mid 0.0 \mid \ldots \mid 1.0 \mid$
Momentum constant: α	$\mid 0.1 \mid \ldots \mid 0.9 \mid$
Activation function	$\mid Tanh \mid Logistic \mid Gaussian \mid$

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.

4.2. Prediction by Neural Network

Artificial neural networks have emerged as excellent tools for deriving data oriented models because of their inherent characteristic of plasticity that permits the adaptation of the learning task when data is provided. In this regards, the use of an ANN is adequate to derive the forecasting model proposed. In addition to plasticity, ANN also present generalization and fault tolerance characteristics that are fundamental for systems that depend on observational data that may be incomplete and slightly different from the data used to derive the models [1].

The problem worked in this paper is the derivation of an ANN based model for climate forecasting using re-analysis data. Among the different ANN models available in the literature, an MLP was chosen because of its appropriateness for supervised training under the existence of data. The available data consist of the history of the atmosphere in a certain period of time [1].

5. CASE STUDY: SEASONAL CLIMATE PRECIPITATION PREDICTION

Climate forecasting is the estimation of the average behavior of the atmosphere for a future period of time (more than one month ahead). For instance, in making a seasonal climate forecast, one may study if the next season (e.g. winter) will colder than the average, or else, if there will be more rain fall than in the previous season. Thus, the objective of climate forecasting is to estimate the statistical properties of the climate in a future period of time [12].

In Meteorology, weather and climate are distinct concepts related to the study of the atmosphere. Weather studies are related to a certain geographic region in a very short time.

This is a very complex task due to the inherent dynamics of the physics of the atmosphere that may directly affect human life on the planet. Climate studies search for understanding the average behavior of the atmosphere in a long period of time by integrating the atmospheric conditions to represent an abstract characterization [12].

Climate prediction centers conduct climate forecasting using models that try to describe the behavior of the physical-chemical conditions of the atmosphere. Such models require high performance computational resources to generate possible future status of the atmosphere in high resolution scales. The outputs of such models and systems are used to support the studies of impacts and vulnerabilities, and to allow of prediction of atmospheric extreme conditions. The models may be local, regional, or global in scope.

5.1. Data used

The data was downloaded from the reanalysis data repository from NCEP/NCAR (National Center for Environmental Prediction[1] / National Center for Atmospheric Research). The data consists of monthly means from January 1999 to December 1999 in North Brazil, within the geographic coordinates (Lat 15S, 10N) to (Lon 75W, 45W) with a spatial resolution of 2.5. The available variables are: zonal wind components at vertical levels 300 hPa, 500 hPa, meridional wind components at vertical levels 300 hPa, 500 hPa, and precipitation.

The training data set was formed with data from January 1999 up to June 1999. This set was used to derive a neural network model. The remaining (July to December 1999) was used to derive the generalization data set. A testing set corresponding to 25% of the training set was randomly created to be used as an anticipated stopping criterion.

6. RESULTS

The following are the computational results obtained with MPCA. The algorithms are implemented in Fortran 90 and computer tests were conducted under Linux operating system, in a 6 nodes cluster which 2 processors Pentium III de 1.26 GHz per node.

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: 6 particles; 6 processors; 500 iterations. The stopping criterion used was the maximum number of evaluations of the objective function.

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 realization (performed by an expert). As can be seen, the results are in a good agreement. The procedure of an automatic method to identify a good configuration for the ANN does not require the supervision of an expert.

Figure 3. ANN results, with configuration determined by MPCA and Empirical processes.

[1] `http://www.ncep.noaa.gov/`

Table 3. Configuration Architecture

Parameter	ANN MLP - MPCA	ANN - Empirical
Hidden Layers	2	1
Neuron in 1 hidden layer	9	6
Neuron in 2 hidden layer	8	0
Neuron in 3 hidden layer	0	0
Learning rate η	0.53	0.4
Constant momentum α	0.20	0.6
Activation function	$Tanh$	$Logistic$
Mean Square Error	0.000076	0.000048

7. CONCLUSION

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 seasonal climate predition with focus on precipitation, where reanalysis data from NCEP/NCAR was used for training and generalization tests. 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 could 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. Author H.F. de Campos Velho want also to thank the FINEP from the support to the CLIMARS project.

8. REFERENCES

[1] Anochi J. A., "Modelos baseados em redes neurais para o estudo de padrões climáticos sazonais a partir de dados tratados com a teoria dos conjuntos aproximativos". *M.Sc. dissertation on Applied Computing – Intituto Nacional de Pesquisas Espaciais*, August-2010.

[2] 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.

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

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

[11] 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.

[12] Vianello R. L., Alves A. R., "Meteorologia básica e aplicações". *Universidade Federal de Viosa* (UFV), Viçosa, 2006.

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