Optimization of Feedforward Neural Network by Multiple Particle Collision Algorithm

Juliana A Anochi
Applied Computing Graduate Program
National Institute for Space Research
São José do Campos, SP, Brazil
Email: juliana.anochi@lac.inpe.br

Haroldo F. de Campos Velho
Laboratory Computing and Applied Mathematics
National Institute for Space Research
São José do Campos, SP, Brazil
Email: haroldo@lac.inpe.br

***Abstract*—Optimization of neural network topology, weights and neuron activation functions for given data set and problem is not an easy task. In this article, a technique for automatic configuration of parameters topology for feedforward artificial neural networks (ANN) is presented. The determination of optimal parameters is formulated as an optimization problem, solved with the use of meta-heuristic Multiple Particle Collision Algorithm (MPCA). The self-configuring networks are applied to predict the mesoscale climate for the precipitation field. The results obtained from the neural network using the method of data reduction by the Theory of Rough Sets and the self-configuring network by MPCA were compared.**

I. Introduction

Artificial Neural Networks (ANNs) are computational techniques that present a mathematical model inspired by the neural structure of biological organisms, acquiring knowledge through experience which have been applied widely in several tasks such as control, prediction of weather and climate, data assimilation, optimization, image processing and others. ANNs have emerged as excellent tools for deriving data oriented models, due to their inherent characteristic of plasticity that permits the adaptation of the learning task when data is provided. In addition to plasticity, ANNs 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.

The use of ANNs to solve real problems always includes the selecting of appropriated network topology. The later issue has been the subject of intensive research in recent years, because we are looking for an optimal topology to solve a specific problem. This is a complex task, and usually requires a great effort by an expert, identifying the best parameter set, and it is necessary a previous knowledge about the problem to be treated.

However, increased attention has been especially directed to finding the best topology. This is justified not only by the fact that it is directly associated with the models performance but also because there is no theoretical basis of how we find the best topology among the many possible options. In practice, this problem is usually solved in part by using empirical or statistical methods that are used to study the effect of an ANNs internal parameters and choose appropriate values for them based on repetitive trial and error method.

Nevertheless, 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 topology specification, such as: (a) Pruning (Hinton, 1989) [1] makes adjustment of neural network by modifying its structure, the training begin with an oversized topology and the weights are eliminated until the capacity of generalization can be increased; (b) Weight Decay suggested by (Hinton, 1989) [1]: the algorithm is similar to the Pruning, where the cost function and weight vector are modified; (c) Early Stopping proposed by (Weigend, 1990) [2], the scheme performs the early interruption of training, without changing the ANN topology, similarly; (d) Cross Validation proposed by (Stone, 1978) [3]: it is known to improve the generalization, where the data set is separated in two sets: training data and validation data. However, all these algorithms still suffer from slow convergence. In addition, these are based on gradient techniques and can easily stick at a local minimum.

The identification of the best configuration for a given ANNs can be considered as an optimization problem, where each point in the search space represents a potential ANN with different topology and weights. Hence, many methods were presented to tackle this problem.

Zhihong et al. [4] proposes training algorithm for a class of single-hidden layer feedforward neural networks (SLFNs) with linear nodes and an input tapped-delay line memory. The authors in order to remove the effects of the input disturbances, the input weights of the SLFN are assigned such that the hidden layer of the SLFN performs as a pre-processor, and the output weights are then trained to minimize the weighted sum of the output error squares as well as the weighted sum of the output weight squares.

Chaohua et al. [5] detailed the seeker optimization algorithm (SOA), a population-based heuristic stochastic search algorithm, which is based on the concept of simulating the act of human searching for tuning the structures and parameters of ANNs. In the SOA, the search direction is determined by seekers egotistic behavior, altruistic behavior and pro-activeness behavior, while step length is given by uncertainty reasoning behavior.

Valdez et al. [6] used a hybrid approach for optimization combining Particle Swarm Optimization (PSO) and Genetic Algorithms (GAs). This optimization method combines the advantages of PSO and GA to provide an improved FPSO +

978-1-4799-5375-2/14/$31.00 ©2014 IEEE

FGA hybrid method using Fuzzy Logic to to adjust parameters.

In [7] the authors proposed a hybrid method based on particle swarm optimization for designing ensemble neural networks with fuzzy aggregation of responses to forecast complex time series.

This paper deals with self-configuration using a new meta-heuristic called the Multiple Particle Collision Algorithm (MPCA) to compute the optimal topology for an ANN. The cost function has two terms: a square difference between ANN output and the target data for two data set: learning process, and the generalization, and a penalty term used to evaluate the complexity for the new network topology at each iteration. The concept of network complexity is associated to the number of neurons and the number of iterations in the training phase.

Additionally, this work proposes a methodology for developing an empirical model of monthly climate prediction precipitation field, from reanalysis of historical data, the database National Oceanic & Atmospheric Administration (NOAA).

In this same theme, Anochi et al. [8] presented an approach that deals with the use of artificial neural networks models in the derivation of climatological models in prediction seasonal scale, from reanalysis of data processed by a data mining method based on the theory Rough Sets to extract relevant information from this data to perform the seasonal forecast throughout the Brazilian territory, from a reduced set of data.

In this paper, we compare the results obtained from the neural network using the method of data reduction by the Theory of Rough Sets [8] and self-configuring network by MPCA.

II. Configuring the Supervised Networks by Metaheuristics

Currently, one of the main topics of research for supervised neural network is the search and definition of a topology optimal or nearly optimal.

There is not a clear indication of how we find the best topology among the many possible choices. Further, we not even know in advance the correct network topology to be applied. In practice, this problem is usually solved in part by using empirical methods based on repetitive trial and error method. Where the configuration is determined during preliminary tests with different topologies, modifying network parameters, until satisfactory results are obtained. This configuration consists of an iterative process where the specialist changes the values of model parameters for each trial and compared the results with the observed values until you reach a set of parameters for which, in their view, the model results are the most suitable.

The problem to identify the best configuration of the supervised ANN can be treated as an optimization problem. The goal is to find the optimum value, which represents the best combination of variables for the ANN topology, and the definition of the set of weights. This self-configured ANN is determined as the minimization of cost function defined by Equation 1.

The main advantage in using an automatic procedure to configure an ANN is the ability to define a topology near-optimal ANN, without needing the help of an experts on the ANN approach and/or the application. Such approach avoids this time consuming and tiring process of trial and error to find the optimal neural network topology.

For the problem of construction of climate prediction model, as proposed in this paper, the networks multilayer perceptron and recurrent Elman are used.

The objective function used in this study is the square difference between the target values and the ANN output. The latter factor is expressed by equation [9]:

$$f_{\text{obj}} = penalty \times \left(\frac{\rho_1 \times E_{\text{train}} + \rho_2 \times E_{\text{gen}}}{\rho_1 + \rho_2} \right) \quad (1)$$

where, $\rho_1 = 1$ and $\rho_2 = 0.1$ the same values proposed by [9]. These are adjustment factors that modify the attributes relevant to the training (E_{train}) and generalization (E_{gen}) errors.

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 factor *penalty* is applied to compute the ANN with the lowest complexity possible. The computational complexity of a supervised ANN topology can be defined as the total number of neurons and the epochs number present in its structure. To this purpose, a penalty factor was developed to determines the influence of the ANN architecture on the objective function values.

$$penalty = C_1 \left(\varepsilon^{neurons}\right)^2 \times C_2 \left(epochs\right) + 1 \quad (2)$$

where $C_1 = 1$ and $C_2 = 0.1$ are adjustment factors that modify the attributes relevant. The $\varepsilon^{neurons}$ is the neurons total number, usually denoted by an exponential, and the $epochs$ represents the number of training epochs or cycles.

In an optimization problem, such penalty is incorporated to the objective function in such a way to constraint the universe of possible ANN architectures. It penalizes larger and complex ANN architectures with too many neurons in the hidden layers or that need too much CPU time (learning time).

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

TABLE I. Parameters to define a network topology.

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

The goal function (equation 1) is solved by a metaheuristic. This paper presents the use of MPCA for automatic configuration of a neural network topology, minimizing the cost function, producing a configuration of the ANN with the best possible performance.

A. Multiple particle collision algorithm

The Multiple Particle Collision Algorithm was developed by Luz [10], inspired the canonical Particle Collision Algorithm (PCA) [11], [12], [13] but a new characteristic is introduced: the use of several particles, instead of only one particle to act over the search space. In both algorithms are greatly inspired by two physical behaviours, namely absorption and scattering, that occurs inside a nuclear reactor. They are similarities with basic characteristics of Simulated Annealing [14], [15].

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 or cannot be accepted. If the new solution is not accepted, the particle can be send to a different location of the search space, giving the algorithm the capability of escaping a local optimum, this scheme is inspired on the scaterring (Scaterring{.}). If a new solution is better than the previous one, this new solution is absorbed (Absorption.}). The exploration around closer positions is guaranteed by using the functions Perturbation{.} and Small-Perturbation{.} [16], [10].

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. Luz et al. [16] have showed which the MPCA is able to computing good solutions were which PCA cannot found a correct answer.

The MPCA is intended to be implemented using Message Passing Interface (MPI) libraries in a multiprocessor topology with distributed memory.

The pseudo-code for the MPCA is presented by Table II for more detail see [10].

TABLE II. PSEUDO-CODE FOR MPCA.

```
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
```

The MPCA is part of a class of algorithms inspired by natural phenomena. While the PCA is a trajectory-based method, the MPCA is a population-based method, that starts from a set of initial solutions (initial population) and try to find a better solution by change elements of such population.

Exploration function (Exploration), small stochastic perturbation solution are applied using the function (Small-Pertubation) to perform a small local exploration in order to verify the existence of a possible solution even better in a given neighborhood as shown in table III.

TABLE III. EXPLORATION.

```
Exploration(.)
    For n = 0 of # iterations
     Small-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
    End-if
Return
```

The Table IV shows the procedure scattering (Scattering) a stochastic scheme that tries to avoid arrest of the algorithm in a region of space in a local optimum.

TABLE IV. SCATTERING

```
Scattering(.)
    Pscattering = 1 - Fitness(New-Config) / Best-Fitness
    If Pscattering > Random(0,1)
        Old-Config = Random solution
    Else
        Exploration()
    End-if
Return
```

III. REDUCTION OF METEOROLOGICAL DATA BY ROUGH SETS THEORY

Rough Sets Theory (RST) was proposed in 1982 by Zdzislaw Pawlak as a mathematical theory to treat uncertain and imprecise information, by deriving approximations of a data set [18]. Roguh Sets are based on the similarities among objects measured by an indiscernibility relation, which establish that a set of objects are similar (indiscernible) if they hold the same values for all of their attributes. The indiscernibility relation allows the derivation of subsets of variables that maintain the existing information by considering the most relevant attributes of the data, thus it reduces possible redundancy inherent to the data.

The fundamental feature behind Rough Set Theory is the approximation of lower and upper spaces of a set, the approximation of spaces being the formal classification of knowledge regarding the interest domain. Another an important feature the theory RST is the possibility to derive decision rules as a result of the quantitative and qualitative processing of the uncertainty in the available data [19].

Rough sets have been proposed for a very wide variety of applications. In particular, the rough set approach seems to be important for Artificial Intelligence, especially in machine learning, knowledge discovery, data mining, expert systems, approximate reasoning and pattern recognition.

In this paper, the theory RST was used to extract relevant information from meteorological data to perform climate prediction, from a reduced data set.

A. Information Systems

Rough Sets Theory uses the concept of Information Systems (IS) in which the available data are represented in a table in which the objects are displayed in the rows and the attributes in the columns [20]. Formally, an information system is composed of a finite non-empty set U (Universe) of objects and a finite non-empty set A of attributes, $IS = (U, A)$, so that, for each $a \in A$, $a : U \rightarrow V_a$. The set V_a is the set of values of a, that is, the domain of a.

A Decision System is an IS augmented with a decision attribute $d \notin A$. Formally, $DS = (U, A\{d\})$, where $d \notin A$ is the decision attribute [20]. A DS may be deterministic or non-deterministic. A deterministic DS uniquely describes the decisions and actions to be realized when some conditions are satisfied. In the non-deterministic DS the decisions are not determined by the conditions.

B. Indiscernibility Relation

The indiscernibility relation is used as a measure of similarity among objects. Thus, a set of objects with the same attributes are indiscernible if only if their attributes hold the same values from their corresponding domains. This is a equivalence relation that may be used to treat problems as redundancy of attributes or the existence of irrelevant attributes in the data assigned to only one representative of a class.

$$IND_A(B) = \left\{(x, x') \in U^2 | \forall a \in B, a\left(x\right) = a\left(x'\right)\right\} \tag{3}$$

C. Attribute Reduction

The reduction process is identify equivalence classes, i.e. objects that are indiscernible using the available attributes. Savings are to be made since only one element of the equivalence class is needed to represent the entire class. The other dimension in reduction is to keep only those attributes that preserve the indiscernibility relation and, consequently, set approximation [20].

The attribute reduction procedure is performed by the discernibility function $f_A(B)$ derived from the discernibility matrix which is a symmetric matrix constructed by comparing the attribute values that discern the objects. The attribute representing discernible values are inserted into the matrix. Each entry in the matrix consists of a set of attributes that distinguish a pair of objects x_i and x_j expressed by:

$$M_{i,j} = \{a \in B | a\left(x_i\right) \neq a\left(x_j\right)\} \tag{4}$$

where $1 \leq i, j \leq n$ and $n = |U/IND_A(B)|$

The discernibility function $f_A(B)$ for an information system B is constructed by concatenating the subsets $M_{i,j} = \{a^* | a \in M_{i,j}\}$ of attributes in each position of the discernibility matrix M, through a Boolean function of m variables that correspond to attributes $a_1, ..., a_m$. The function determines the minimum set of attributes that distinguish any class among the existing classes [18] and it is defined as:

$$f_A\left(a_1^*, \ldots a_m^*\right) = \bigwedge \left\{\bigvee M_{i,j}^* | 1 \leq j \leq i \leq n, M_{i,j} \neq \oslash\right\} \tag{5}$$

IV. Application: climate precipitation prediction

Climate prediction 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 evaluate if the next season (e.g. winter) will be colder (or more than warm) than the climatological average, or else, if there will be more (or less) rain fall than in the previous season. Thus, the objective of climate prediction is to estimate the statistical properties of the climate in a future period of time [21].

Climate prediction centers conduct climate prediction 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.

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.

A. Description of data

Two study areas were selected to test the adequacy of the proposed methodology to derive forecasting models from data: Southeast Brazil (Lat 25°S, 15°S) to (Lon 52°W, 40°W) and South Brazil (Lat 35°S, 22.5°S) to (Lon 60°W, 45°W).

The data consists of monthly means from January 2000 to December 2009. The data was downloaded from the reanalysis data repository from National Center for Environmental Prediction & National Center for Atmospheric Research (NCEP/NCAR). The world is divided into parallels (latitudes) and meridians (longitude). The global data reanalysis grid uses a set with horizontal resolution of 2.5° × 2.5° (latitude × longitude) [22].

During the training phase, the capacity of generalization is assessed using a set of validation data. The data-set is divided into a subset for training and a subset of validation. The subset for training is used to find the connection weights. The training session is interrupted periodically after a certain number of epochs, and the network is tested with the subset of validation (cross validation process). These testing cycles are repeated until convergence or the maximum number of epochs to be reached. When the validation error tends to increase, characterized by a worse performance during the generalization, the training process is finished.

The training data subset was formed with data from January 2000 up to December 2006. This subset was used to derive a neural network model. The period from January 2007 to December 2008 was used for the validation. The generalization test subset corresponds to the period from January 2009 up to January 2009.

The available variables are presented in Table V.

TABLE V. AVAILABLE VARIABLES

	$Variables$	$Descriptions$
1	$airt$	$Air\ temperature$
2	$u300$	$Zonal\ WindComponents\ 300hPa$
3	$u500$	$Zonal\ Wind\ Components\ 500hPa$
4	$u850$	$Zonal\ Wind\ Components\ 850hPa$
5	$v300$	$Meridional\ Wind\ Components\ 300hPa$
6	$v500$	$Meridional\ Wind\ Components\ 500hPa$
7	$v850$	$Meridional\ Wind\ Components\ 850hPa$
8	$spres$	$Surface\ pressure$
9	$shum$	$Specific\ humidity$
10	$prec$	$Precipitation$

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

$$E_{gen} = \frac{1}{N} \sum_{k=1}^{N} (y_k - d_k)^2 \tag{6}$$

where N is the number of patterns in the data set, y_k is the true observational value, and d_k is the estimation computed by the neural model.

For the visualization of the results we used the GrADS software (Grid Analysis and Display System). This software is used for visualization and data analysis grid points, which provides an integrated environment for access, manipulation and display of data. Software GrADS is a available for free download at [23].

V. EXPERIMENTS AND RESULTS

The seasonal climate prediction model developed in this paper has the objective to develop estimates of the statistical properties of some future climate state, which can be for a month or a season. For example, assuming the initial conditions as model receives data regarding the autumn season, as output will have a forecast for the season ahead, which in this case is the estimate of precipitation for the summer season.

Four models of neural networks were constructed following the same methodology presented in [8]: two models are generated from the network recurrent Elman using all available variables in the database: (i) ANN Elman configured by expert and (ii) ANN Elman self-configuring by MPCA. The other two models, containing a reduced number of variables indicated by RST, also using: (iii) ANN Elman configured by expert and (iv) ANN Elman self-configuring by MPCA.

A. Results ANN self-configuring by MPCA

Parameters used in the MPCA: 6 particles – multi-processing machine: one particle per processor, 10 iterations (a scheme used to explore a better solution around the new particle location). The stopping criterion is the maximum number of objective function evaluations (for the worked exemple: 500). The numerical experiment was performed with synthetic observational data. The MPCA was applied to optimize the parameters of ANN with only one hidden layer: number of neurons in the hidden layer, activation function, learning rate, and momentum.

The MPCA is used to generate a set of candidate solutions that correspond to an ANN topology. 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 (Equation 1 and 2), up dating the parameters for the ANN. This process is repeated until an optimal value (minimum) for the objective function is found.

Table VI shows the settings obtained by using the optimization algorithm of the following networks: ANN1 is the network recurrent Elman that was design by MPCA trained with all available variables in the database. ANN2 is the result of recurrent network Elman, had its topology configured using the MPCA algorithm. The training was performed with a reduced set of variables indicated by RST and ANN3 is the result of network recurrent Elman configured by an expert.

TABLE VI. CONFIGURATION TOPOLOGY THE SUPERVISED ANN

Parameters	ANN1	ANN2	ANN3
intermediate layer	1	1	1
neuron in first layer	25	29	6
learning rate η	0.83	0.85	0.4
momentum α	0.63	0.65	0.6
activation function	$Logistics$	$Logistics$	$Logistics$

B. Results the rough set reducts

In the dimensionality reduction process the relevant attributes are those that mostly occur in the data, in terms of the indiscernibility relation. For the attribute reduction process the training data set previously mentioned was first discretized and then submitted to the reduction algorithm for selection of the relevant attributes chosen as those with a presence greater than 70% in the discernibility function. The resulting reducts for both study areas are presented in Table VII.

Table VII shows that only 7 variables out of 10, were considered for South and only 5 variables out of 10 were considered for Southeast. It is to be noticed that the variables that form the reducts have a presence greater than 70% in the dicernibility function. These reducts were then used to derive the second forecasting models for both study areas.

TABLE VII. RESULTING REDUCTS

Reduct for South		**Reduct for Southeast**	
$Variable$	$Presence\%$	$Variable$	$Presence\%$
$airt$	77%	$airt$	78%
$u300$	79%	$u850$	79%
$u850$	80%	$v300$	84%
$v300$	77%	$v500$	80%
$v500$	79%	$spres$	89%
$v850$	70%		
$spres$	78%		

Table VIII presents the mean square error as a performance measure of the result of climate prediction by using Elman recurrent network, which was trained with all the variables available in the database, with two approaches to training: first one, the topology was configured by an expert, and the second, the topology is self-configured by the use of metaheuristic MPCA.

Table IX presents the mean square error as a performance measure for Elman recurrent network trained with the reduced set by rough set theory for both areas.

Figure 1 shows the results obtained for climate prediction in two area: South and Southeast of Brazil using the network

TABLE VIII. MEAN SQUARE ERROR FOR ANN TRAINED WITH ALL THE AVAILABLE VARIABLES

ANN with all the variables	*South*	*Southeast*
Elman configured by expert	0.059	0.004
Elman configured by MPCA	0.016	0.036

TABLE IX. MEAN SQUARE ERROR FOR ANN TRAINED WITH THE REDUCED SET

ANN with the reduced set	*South*	*Southeast*
Elman configured by expert	0.017	0.011
Elman configured by MPCA	0.002	0.038

recurrent Elman. Figure 1(a) corresponds to the observed precipitation in Southeast in winter 2007; The Figure 1(b) corresponds to the result produced by the network Elman trained with all the available variables in the database and the topology was configured by an expert; Figure 1(c) corresponds to the result produced by the network Elman trained with a reduced set of variables processed by the RST and the topology was design by an expert; Figure 1(d) shows the result obtained by the network recurrent Elman that was design by MPCA trained with all available variables in the database; and Figure(e) corresponds to the result produced by the recurrent network Elman trained with the reduced set of variables processed by rough set theory, and the topology was configured using the MPCA algorithm. The same order is followed in the Figure 2. In Figure 1(e), Elman with the reducts and by design MPCA looks the most simliar to the observation Figure 1(a).

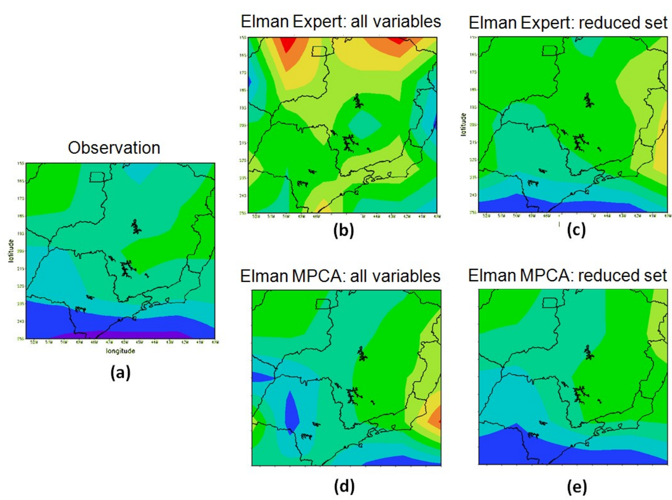

Fig. 1. The analysis region: Southeast of Brazil. (a) observed precipitation in summer 2007; (b) network Elman trained with all the available variables in the database and the topology was configured by an expert; (c) estimation produced by the Elman trained with the data processed by the RST; (d) estimation produced by the network Elman trained with all available variables in the database and the topology was design by MPCA; and (e) correspond to the estimation produced by the Elman trained with the reducts, and the topology was configured by MPCA.

Figures 3 and 4 shows the error between the observation and the climate prediction executed by Elman network, and the performance using the two strategies. In the first strategy (see Figures 3a and 4a) the networks were trained with all available variables and the second, were trained with the reduced data

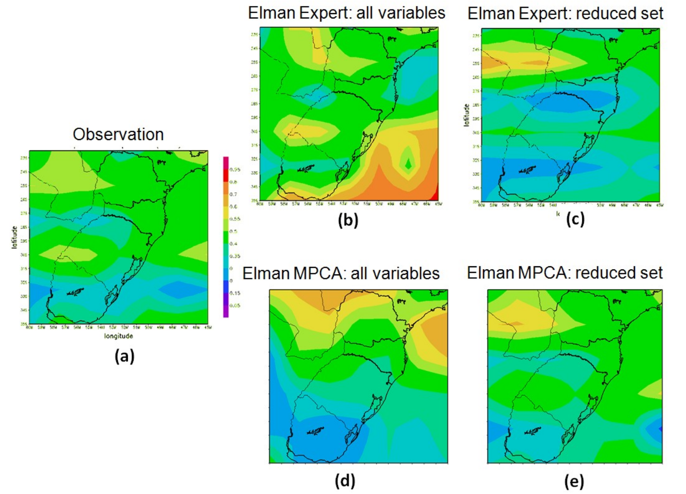

Fig. 2. The analysis region: South of Brazil. (a) observed precipitation in winter 2007; (b) network Elman trained with all the available variables in the database and the topology was configured by an expert; (c) estimation produced by the Elman trained with the data processed by the RST; (d) estimation produced by the network Elman trained with all available variables in the database and the topology was design by MPCA; and (e) correspond to the estimation produced by the Elman trained with the reducts, and the topology was configured by MPCA.

by RST (see Figures 3b and 4b).

In the Figures 3a and 4a (strategy first), the red line corresponds the results for climate prediction obtained by Elman network configured by MPCA, and the blue line are the results obtained by the Elman-Expert. In the Figures 3b and 4b (strategy second), the red line corresponds the results obtained by Elman-MPCA, and the blue line are the results obtained by the Elman-Expert.

From the results, the Elman network configured by MPCA presented the best performance than ANN defined by an expert. Indeed, both ANNs computed by MPCA show a high fidelity with real dynamics.

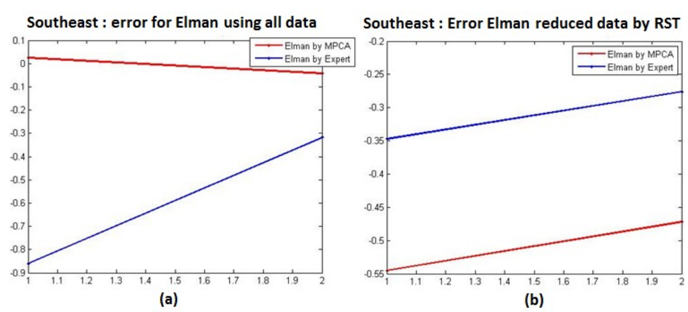

Fig. 3. Difference between observed and the output of Elman network: Southeast of Brazil. (a) Error for Elman network trained with all the available variables in the database; (b) Error Elman network trained with the reduced data processed by the RST.

VI. CONCLUSION

The supervised networks were applied to climate prediction of precipitation field. Climate prediction is an important issue with strong impact for the society, in particular the precipitation field, one of more difficult meteorolical variable

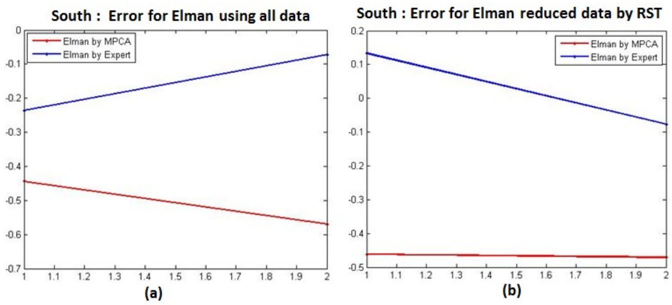

Fig. 4. Difference between observed and the output of neural networks: South of Brazil. (a) Error for Elman network trained with all the available variables in the database; (b) Error Elman network trained with the reduced data processed by the RST.

to be predicted, due to its large variability in space and time. This is a task that has been pursued by humans, for a longer a time.

Anochi and Silva [8], [24] have applied ANN for sasonal climate prediction for precipitation field. The authors demonstrated the efficacy of the proposed method, with estimates of predicted similar to climatological conditions considered as available observations in the database.

However, the configuration of a supervised ANN is not a easy task and usually requires a great effort by the expert mainly to determine the best parameters, and it is necessary a previous knowledge about the problem to be treated. Many real-world problem involves the optimization of several incommensurable and often conflicting objectives. For this reason, optimization the parameters of ANN has become an important topic and, very challenging for researchers from several applied sciences.

The problem to identify the best configuration of a supervised network to the application cited above is formulated as an optimization problem. The stochastic technique MPCA was employed to address the solution of the optimization problem.

The empirical prediction model described in the present study provides the development of future scenarios for supporting the studies of impacts and vulnerability, and it can still enable to display projections of climate extremes of atmospheric state, demanding lower computing power.

The analysis of this results highlights the advantages of using self-configuration approach in the setting for optimal parameters of ANNs models is crucial for convergence and to avoid overfitting, a problem that usually happens in training patterns.

Acknowledgment

The authors would like to thank CNPq and CAPES for the physical and financial support provided to this research.

References

[1] G. E. Hinton, “Connectionist learning procedures,” *Artificial Intelligence*, vol. 40, pp. 185–234, 1989.

[2] A. Weigend and B. Huberman, “Predicting the future: a connectionist approach,” *International Journal of Neural Systems*, vol. 1, no. 3, pp. 193–209, 1990.

[3] M. Stone, “Cross-validation: A review,” *Mathematische Operationforsch Statischen*, vol. 9, pp. 127–139, 1978.

[4] M. Zhihong, L. Kevin, W. Dianhui, C. Zhenwei, and M. Chunyan, “A new robust training algorithm for a class of single-hidden layer feedforward neural networks,” *Neurocomputing*, vol. 74, no. 16, pp. 2491–2501, 2011.

[5] D. Chaohua, C. Weirong, Z. Yunfang, Z. J., and Y. Zhiyu, “Seeker optimization algorithm for tuning the structure and parameters of neural networks,” *Neurocomputing*, vol. 74, no. 6, pp. 876–883, 2011.

[6] O. C. F. Valdez, P. Melin, “Modular neural networks architecture optimization with a new nature inspired method using a fuzzy combination of particle swarm optimization and genetic algorithms,” *Information Sciences*, vol. 270, pp. 143–153, 2014.

[7] O. C. M. Pulido, P. Melin, “Particle swarm optimization of ensemble neural networks with fuzzy aggregation for time series prediction of the mexican stock exchange,” *Information Sciences*, vol. 280, pp. 188–204, 2014.

[8] J. Anochi, “Modelos baseados em redes neurais para o estudo de padres climticos a partir de dados tratados com a teoria dos conjuntos,” Master’s thesis, Instituto Nacional de Pesquisas Espaciais-INPE, São José dos Campos, 2010. [Online]. Available: http://urlib.net/sid.inpe.br/mtc-m19/2010/08.25.12.26

[9] A. Carvalho, M. Ramos, and A. Chaves, “Metaheuristics for the feedforward artificial neural network (ann) architecture optimization problem,” *Neural Comput Applic*, vol. 20, pp. 1273–1284, 2011.

[10] E. F. P. d. Luz, “Parallel metaheuristics in solving inverse problems,” Ph.D. dissertation, National Institute for Space Research, São José dos Campos, 2012-03-08 2012. [Online]. Available: http://urlib.net/sid.inpe.br/mtc-m19/2012/02.22.17.13

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

[12] W. Sacco, C. Lapa, C. Pereira, and H. Filho, “Two stochastic optimization algorithms applied to nuclear reactor core design,” *Progress in Nuclear Energy*, vol. 48, pp. 525–539, 2006.

[13] W. F. Sacco, C. M. F. Lapa, C. M. N. A. Pereira, and H. A. A. A. Filho, “A metropolis algorithm applied to a nuclear power plant auxiliary feedwater system surveillance tests policy optimization,” *Progress in Nuclear Energy*, vol. 50, pp. 15–21, 2008.

[14] N. Metropolis, A. Resenbluth, M. Rosenbluth, and T. E. Teller, A.H, “Equation of state calculations by fast computing machines,” *The Journal of Chemical Physics*, vol. 6, no. 21, pp. 1087–1092, 1953.

[15] S. Kirkpatrick, C. Gelatt, and M. Vecchi, “Optimization by simulated annealing,” *Science*, pp. 671–680, 1983.

[16] E. Luz, J. Becceneri, and H. Velho, “A new multi-particle collision algorithm for otimization in a high-performance environment,” *Journal of Computacional Interdisciplinary Sciences*, vol. 1, pp. 1–7, 2008.

[17] A. Eiben and J. Smith, *Introduction to Evolutionary Computing*, 1st ed. Verlag Berlin: Springer-Verlag, 2003.

[18] Z. Pawlak, “Rough sets,” in *Proceedings of the International Journal of Computer and Information Sciences*, vol. 1, Poland, 1992, pp. 341–356.

[19] J. Komorowski, Z. Pawlak, L. Polkowski, and A. Skowron, “Rough sets: A tutorial,” *Rough fuzzy hybridization: A new trend in decision-making*, pp. 3–98, 1999.

[20] J. Komorowski. and A. Ohrn., “Modelling prognostic power of cardiac tests using rough sets,” *Artificial Intelligence in Medicine*, vol. 15, pp. 167–191, 1999.

[21] R. L. Vianello, *Meteorologia básica e Aplicações*. UFV, 2006.

[22] E. Kalnay, *Atmospheric modeling, data assimilation, and predictability*. Cambridge university press, 2003.

[23] B. Doty. (2009, February) Grid analysis and display system (grads). Maryland: Center for Ocean- Land-Atmosphere Studies (COLA). [Online]. Available: http://grads.iges.org/grads/head.html

[24] J. Anochi and J. Silva, “Using artificial neural networks and rough set theory in the study of seasonal weather patterns,” *Learning and Nonlinear Models*, vol. 7, pp. 83–91, 2009.