

Data Assimilation using NeuroEvolution of Augmenting Topologies

André Grahl Pereira

Computer Science Post-graduate Program – PPGCC
Federal University of Rio Grande do Sul – UFRGS
Porto Alegre, Brazil
andre.grahl.ai@gmail.com

Adriano Petry

Southern Regional Space Research Center – CRS
National Institute for Space Research – INPE
Santa Maria, Brazil
dr.adriano.petry@gmail.com

Abstract— The use of numerical prediction models are essential to modern society. Data assimilation is a technique that aims to increase the prediction accuracy by combining a model output with observational data, resulting in a state that is closer to the true state of the problem. Depending on the size of the model output and the number of observations to assimilate, the combination of these two sources of information may require intensive computing and become a challenge, even for supercomputers used in this type of application. Thus neural networks have been proposed as an alternative to perform high quality data assimilation at lower computational cost. This paper investigates the use of NeuroEvolution of Augmenting Topologies (NEAT) in data assimilation. NEAT is capable of adapting the connections weights and the neural network topology using principles of evolutionary computation in a search for a minimum topology and best performance. In this work, two different models were used for testing: the Lorenz Attractor and Shallow Water model. The experiments compared the results obtained with NEAT and backpropagation neural networks, using as benchmark the Best Linear Unbiased Estimator (BLUE). In the experiment with the Lorenz Attractor, NEAT was able to emulate the data assimilation task with smaller error at lower computational cost. For the Shallow Water model, tested using different grid sizes, it was observed that the errors obtained with both neural networks were small, but NEAT showed high error values. On the other hand, NEAT always gets a topology with significantly fewer operations, and the computational cost difference increases with the grid size.

Keywords- data assimilation; NEAT; neural networks; BLUE; Lorenz attractor; shallow water model;

I. INTRODUCTION

Numerical prediction models are essential to modern society. They are used to simulate and forecast important physical phenomena which present, in many cases, a very complex and non-linear behavior that prevents accurate long-term predictions. The numerical weather prediction, for example, generally depends on accurate representation of the initial state, and a small error in this state can cause large errors in the predicted state [1]. In this context, data assimilation appears as a technique to increase the prediction accuracy of numerical models [2].

Data assimilation is a methodology that offers an efficient way to combine data from a numerical prediction model and

data from observation instruments, resulting in a state that is closer to the true state of the problem. Nowadays, it is not hard to find a lot of observations for a variety of domains where numerical models are available. These observations are obtained from instruments, such as balloons, barometers, radars, satellites and ionosondes, among others. Data assimilation techniques combine these data with a model output, and generate other output state, minimizing errors arising from observational instruments or modeling. Depending on the size of the model output and the number of observations to assimilate, the combination of these two sources of information may require intensive computing and become a challenge even for supercomputers [3], especially when dealing with numerical prediction models of practical interest, like terrestrial climate or space weather. Faced with this challenge, new data assimilation techniques have been proposed, aiming at reducing computational cost while maintaining assimilation quality. Neural networks have been used as an alternative to classic methods of data assimilation.

The use of neural networks for data assimilation is recent. Neural networks were suggested as a possible technique for data assimilation by [4], but the first implementation is described in [5], that proposed the use of a neural network for data assimilation in two non-linear dynamic models: the Henon and Lorenz models in chaotic state. The approach used does not require recalculation of the gain matrix at each time step of assimilation. The results obtained were promising. Gardner and Dorling [6] presented an extended survey on the use of neural networks as an alternative to the traditional methods of statistical modeling. In addition, they presented several recent neural network applications in atmospheric problems. In the work [7] a radial base function neural network was applied for data assimilation in the Lorenz model. The results were better when compared with the standard model of neural network. In [8] the Shallow Water model was introduced as a mean to evaluate the performance of data assimilation approaches using neural networks. In the paper [9] the authors evaluated the performance of two feedforward (multilayer perceptron and radial basis function) and two recurrent (Elman and Jordan) neural networks. The Lorenz system under chaotic regime was used as case study. These four neural networks were trained to reproduce the behavior of a classical data assimilation method using cross-validation scheme. Multilayer perceptron and Elman showed the better results. In the recent work [10], neural

networks were applied to the problem of data assimilation in the global model Simplified Parameterizations, primitive-Equation DYNamics (SPEEDY). The results with neural networks were very close to the results obtained from a classical data assimilation method. The tests showed that the major advantage of using neural network was the better computational performance, with similar quality of output.

With the exception of the paper [9], all works in the literature that used neural networks in data assimilation employed a direct neural network with fixed topology trained by backpropagation. The work [9] showed that it is possible to increase the computing performance with the use of recurrent connections. However, no work investigated the variation of one of the most important characteristics of a neural network: the topology. In [9], the authors used different neural network models, all at fixed topology. The models used were: direct neural network, radial basis function neural network, Elman network and Jordan network, applied to Lorenz models and Shallow Water and trained by backpropagation. The recurrent neural networks in the training phase showed a superior computing performance, because they required a significantly lower number of iterations for training. In addition, they were able to solve the problem of data assimilation with similar quality, but with a smaller number of neurons.

Differently from previous works in literature, this paper investigates the use of NeuroEvolution of Augmenting Topologies (NEAT) [11] in data assimilation. NEAT is capable of adapting the connections weights and the neural network topology using principles of evolutionary computation in a search for a minimum topology and best performance. In this work, two different models are used for testing: the Lorenz Attractor and Shallow Water model. These models were chosen since they are references in the literature for testing new techniques in data assimilation. Besides that, the Lorenz model presents a chaotic behavior and sensitivity to initial conditions, like many physical systems. Shallow Water model is appropriate also to evaluate computational performance, since the size of state can be increased arbitrarily. The experiments compared the results obtained with NEAT and backpropagation neural networks, using as benchmark the Best Linear Unbiased Estimator (BLUE).

This paper is organized as follows. Section 2 shows a brief overview of data assimilation, focusing in the Best Linear Unbiased Estimator method. The next section discusses ideas behind neural networks, giving special attention to NeuroEvolution of Augmenting Topologies. Section 4 is dedicated to depict the two benchmark problems that are used in this paper: Lorenz Attractor and Shallow Water. In Section 5, the experiments and numerical results obtained are shown and evaluated. The paper ends with conclusion and future work.

II. DATA ASSIMILATION

Data assimilation process initiates receiving output data from a mathematical or empirical model, which is usually used to predict the state of a given domain at future time. As data from observational instruments are available, they are assimilated to that state, resulting in a different one, hopefully closer to the real state of the problem. Data assimilation is one

of the typical inverse problems where it is available a data set with insufficient information for obtaining the complete result.

In the formulation of the problem, the representation of a given state is called vector x . The representation of the true state of the problem is x^t . The state vector x^b is a reference or background state, obtained by simulation from a computational model. The vector x^a is the resulting state (or analysis), which is obtained by the data assimilation process. Thus the data assimilation problem can be defined by the relation $x^a = x^b + \delta x$, where x^a should be as close as possible to x^t , and δx is defined as the analysis increment.

The available observations are grouped into the vector y . The model and observation spaces and dimensions are different. To compare observation vector y with the state vector x^b , a transformation function (H operator) of the model space to observation space is needed. The difference between the observation vector and the reference model vector, i. e., $y - H(x^b)$ is called the innovation vector and, when applied to a weight matrix, generates the analysis increment δx [12].

A. Best Linear Unbiased Estimator (BLUE)

BLUE is a fundamental data assimilation method, which takes in account the errors from modeling and observational instruments to generate the best possible analysis. To represent the uncertainty about the model and the observations, the error models are assumed to follow a Gaussian probability density function (PDF). This method is defined in (1) and (2), where the matrix K represents the weight matrix or gain function that is applied to the innovation vector.

$$x^a = x^b + K(y - H(x^b)) \quad (1)$$

$$K = BH^T(HB^T - R)^{-1} \quad (2)$$

Since the PDFs of the errors are assumed to be Gaussian, then x^a is also the minimum variance estimation of x^t . B and R are the covariance matrices of model and observation errors and are calculated using (4) and (6), where e_b (3) and e_o (5) are the background and observation error vectors. For a sufficiently large number of error vectors, it is expected that the statistical data converge to values that depend only on the physical processes responsible for those errors.

$$e_b = x^b - x^t \quad (3)$$

$$B = \overline{(e_b - \bar{e}_b)(e_b - \bar{e}_b)^T} \quad (4)$$

$$e_o = y - H(x^b) \quad (5)$$

$$R = \overline{(e_o - \bar{e}_o)(e_o - \bar{e}_o)^T} \quad (6)$$

The computational challenge of BLUE is basically the matrix operations, and mainly the matrix inversion needed to calculate the gain function K (2), since the matrix dimensions may achieve thousands of values. The neural network could be a solution to reproduce a similar behavior of the gain function, with the advantage of significant reduction of computational cost.

III. NEURAL NETWORKS

Artificial neural networks are mathematical models inspired by functional or structural aspects of biological neural networks. They are applied in modeling complex patterns between input and output or to find patterns in data. A neural network consists of an interconnected group of neurons processing information using a connectionist approach.

The topology of a neural network has a great influence on its behavior and defines how neurons connect to each other to form the network of neurons. Although a fully connected neural network with one hidden layer can approximate any continuous function, the work of Stanley and Miikkulainen [11] shows that the choice of suitable topology can significantly improve its performance. Unlike learning techniques, there isn't a sufficiently general and efficient methodology to determine the best way that neurons must be connected. One answer to this problem is the NeuroEvolution of Augmenting Topologies (NEAT).

A. NeuroEvolution of Augmenting Topologies

NEAT is a method that evolves weights and topology of a neural network based on three principles: application of a permutation operator, structural innovation using speciation, and incremental growth from minimal structure [11]. When compared to other methods of neuro-evolution, NEAT provides the best results in problems such as XOR and classic inverted pendulum.

The NEAT method is classified as a Topology and Weight Evolving Artificial Neural Networks (TWEANN), which are used in combination with genetic algorithms and neural networks in order to evolve weights and topologies. One challenge is to define how the phenotype (neural network) will be represented in the form of genotype, since the genetic algorithm operates only on the genotype. There are two main categories of coding, direct and indirect. The direct coding stores all the information necessary for the construction of phenotype. The simplest form of direct encoding of a neural network in the form of a chromosome is through a list of genes. Fixing the topology of the neural network, each connection is mapped to a position on the chromosome. Another form of direct coding is the adjacency matrix. Both have as main limitation the need to establish a fixed topology. Indirect encoding happens when only the rules of construction are stored in the genotype, so the phenotype can be built from these rules. The indirect encoding allows a more compact representation compared to direct coding, since each connection to each neuron is not specified in the genome.

The encoding used in NEAT is direct encoding, where two lists are used: one storing neurons found in the neural network and other storing all connections. This encoding does not impose any restriction on the topology of the neural network, which can result in arbitrary structure or size. In the list of neurons each node has a unique identifier, and type: input, hidden, or output. In the list of connections, the identifier stores information about where the connection initiates and finishes, synaptic weight, and the identification of innovation. This identification is assigned to each connection that arises from mutation.

Another challenge is known as the permutation problem, because two neural networks that solve the same problem may have different genic representations. Thus, genes can be lost during the permutation, making the resulting children less efficient than the parents. The solution proposed in NEAT is the use of a permutation operator that considers the track record of innovation. The genotypes of the parents are aligned through the historical record, and the genes can be classified into three categories: homologous, disjoint or exceeded. Homologous genes are those that exist in both parents. Disjoint genes exist in only one parent and exceeded genes exist in only one parent that has greater historical record. At the end of the gene alignment, homologous genes are inherited randomly. Exceeded and disjoint genes are inherited only from the parent with better performance. This solution minimizes the problem in permutation.

Finally another important feature of NEAT is complexification. NEAT is different from other neuroevolutional methods, where the first generation is initialized randomly. It allows great genetic variability for the population, but makes it difficult to guarantee that the minimal topology is found. For NEAT, the first generation starts with minimum topology, where the input layer is connected to the output layer, and the complexity will occur in each generation through mutation and permutation operators.

IV. BENCHMARK PROBLEMS

Numerical models are basically composed by computer simulations that have an initial condition, and evolve over time using physics concepts of the domain or empirics. They can be used to evaluate and predict future states, based on its equations and a previous state. Two numerical models are used in this work: the Shallow Water model and the Lorenz Attractor. These models were chosen because they are references in the literature since they were extensively used on similar works. Lorenz model also presents a chaotic behavior and sensibility to initial conditions, and in Shallow Water model the size of state can be varied arbitrarily, allowing a more detailed evaluation of computational performance.

A. Lorenz Attractor

The Lorenz attractor [13] was first introduced by Edward Lorenz in 1963. The simplified equations were derived from squeegees convection occurring in the atmosphere, that shows how the chaotic state of a dynamical system evolves over time, in a complex and not repetitive pattern. It is a nonlinear system, which presents three-dimensional deterministic chaotic behavior. The system consists of three coupled first order ordinary differential equations: (7), (8) and (9).

$$\frac{dx}{dt} = -\sigma x - y \quad (7)$$

$$\frac{dy}{dt} = -\rho x - y - xz \quad (8)$$

$$\frac{dz}{dt} = xy - \beta z \quad (9)$$

σ , ρ and β are the model parameters, and variables x , y and z have spatial interpretation, as shown in fig. 1. The Lorenz

system is often used to test data assimilation techniques, since it is a simple dynamical system with chaotic behavior and easy to implement.

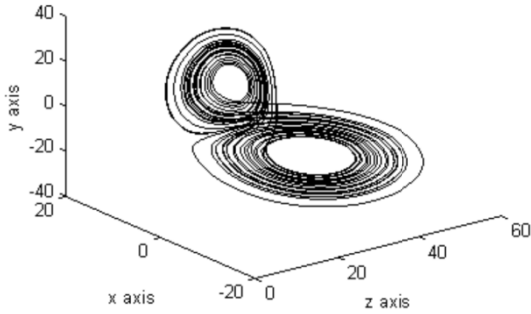


Figure 1. Lorenz attractor representation. [14]

B. Shallow Water

The Shallow Water equations, also called Saint Venant equations [2] are a set of hyperbolic partial differential equations that describe the flow of a fluid surface. They are suitable for flows with a free surface and shallow, as shown in fig.2, and can be used to describe the behavior of a lake or a river. The model provides the time evolution of the water level and horizontal velocity. A simplified expression of the model is shown in (10), (11) and (12).

$$\partial_t h + \partial_x(hu) + \partial_y(hv) = 0 \quad (10)$$

$$\partial_t(hu) + \partial_x(huu) + \partial_y(huv) + \frac{1}{2}g\partial_x h^2 = 0 \quad (11)$$

$$\partial_t(hv) + \partial_x(huv) + \partial_y(hvv) + \frac{1}{2}g\partial_y h^2 = 0 \quad (12)$$

The model can be used for testing data assimilation techniques, since it presents spatial and temporal continuity.

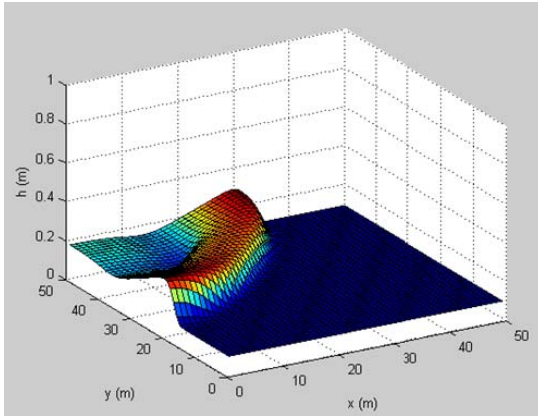


Figure 2. Images with the evolution of shallow water model. [15]

V. EXPERIMENTS AND RESULTS

The experiments compared the results obtained with NEAT and backpropagation neural networks, using as benchmark the Best Linear Unbiased Estimator (BLUE). The prototype for the experiments was developed in C# programming language and used extensively design patterns. It is composed by six main

packages, named Prototype, Experiments, Models, Methods, Dataset and Neural Networks. Simulations with NEAT were implemented using the library SharpNEAT [16]. Simulations with neural network trained by backpropagation were implemented using the framework Aforge.Net [17]. For implementation of BLUE method a large number of matrix operations and algorithms for linear algebra were necessary, so the library Math.NET Numerics [18] was used to meet this need.

A. Test Case: Lorenz Attractor

To enable a realistic comparison with the work done by [9] and [2] the same parameters and methods were used. The Lorenz model was integrated with $\Delta t = 10^{-3}$, and initial conditions $X(0) = 1.50887$, $Y(0) = -1.53127$, $Z(0) = 25.46091$, $\sigma = 10.0$, $r = 8.0/3.0$ and $b = 28.0$. Those parameters guarantee that the system dynamics follow a chaotic trajectory.

Neural networks were trained with a total of 2000 examples, of which 400 examples were used in cross-validation. These values were empirically determined in the works [9] and [2], considering the sufficient number of examples for the neural network learning process. The 400 examples used in cross-validation are considered, according to [9], sufficient to graphically analyze whether the neural network is able to generalize. The best generalization set was defined as 4000 epochs of training. After training and validation, neural networks are tested with prediction of 10^6 steps. This long-term forecast was employed by [9] to verify the generalization capability of neural networks with recurrent connections. The fixed topology of backpropagation neural networks were specified according to the literature, and composed by six neurons in the hidden layer and three in the output layer. The errors of training (ET), cross-validation (EV) and prediction (EP) are given by (13), (14) and (15), respectively.

$$ET = \frac{1}{2000} \sum_{i=1}^{2000} \frac{1}{2} \sqrt{(x_i^{target} - x_i^{nn})^2} \quad (13)$$

$$EV = \frac{1}{400} \sum_{i=1}^{400} \sqrt{(x_i^{target} - x_i^{nn})^2} \quad (14)$$

$$EP = \frac{1}{10^6} \sum_{i=1}^{10^6} \sqrt{(x_i^{target} - x_i^{nn})^2} \quad (15)$$

Table 1 shows the errors of training, cross-validation and prediction obtained for the Lorenz Attractor using backpropagation and NEAT neural networks emulating BLUE method.

TABLE I. ERRORS OBTAINED FOR LORENZ ATTRACTOR EXPERIMENT

	Backpropagation	NEAT
Training	2.92E-05	1.48E-05
Cross-validation	5.38E-04	4.87E-04
Prediction	2.18E-04	1.99E-04

The analysis of table 1 shows that both neural networks were effective and can provide similar results to BLUE method. The error values were comparable, yet NEAT method presented the lowest error for the three data sets.

In terms of computational cost, since backpropagation neural network have fixed topology, a total of 63 multiplication operations are necessary for each output. But the neural network trained by NEAT found a minimal topology of three neurons in the hidden layer and three in the output layer, with a total of 13 connections, which implies directly in only 13 multiplication operations for each output of that neural network.

Thus one can observe that NEAT method was able to solve the proposed problem with a significantly lower computational cost, and still with less error rate for all training sets.

B. Test Case: Shallow Water

For Shallow Water model, we considered four test cases with grid sizes of 5x5, 10x10, 15x15 and 20x20, corresponding respectively to states with 25, 100, 225 and 400 variables. The training process was exactly the same used in the experiment of Lorenz Attractor. Neural networks were trained with 2000 examples, of which 400 examples were used in cross-validation. The best generalization set was defined as 4000 epochs of training. After training and validation, neural networks are tested with prediction 10^6 steps. The neural network with fixed topology trained by backpropagation was specified according to the best performance on the validation set. Considering the softness of the Shallow Water model, the neural network had the best performance with no neuron in the hidden layer, thus obtaining a simple neural network with only the output layer.

Tables 2 to 5 summarize the error results obtained at different grid sizes. Differently from Lorenz experiment, backpropagation neural network always got the lowest error value. For the training set this neural network was far superior from neural network trained by the NEAT method. However, for the validation and prediction sets, the difference between the neural networks was significantly reduced, showing that NEAT is able to generalize effectively, what is essential to use it as an alternative to traditional data assimilation techniques.

TABLE II. ERRORS OBTAINED FOR SHALLOW WATER MODEL EXPERIMENT USING GRID SIZE OF 5X5

	Backpropagation	NEAT
Training	2.94E-05	2.52E-02
Cross-validation	3.75E-03	4.02E-02
Prediction	6.57E-02	9.66E-02

TABLE III. ERRORS OBTAINED FOR SHALLOW WATER MODEL EXPERIMENT USING GRID SIZE OF 10X10

	Backpropagation	NEAT
Training	4.14E-05	1.00E-02
Cross-validation	1.41E-02	2.97E-02
Prediction	2.97E-02	5.21E-02

TABLE IV. ERRORS OBTAINED FOR SHALLOW WATER MODEL EXPERIMENT USING GRID SIZE OF 15X15

	Backpropagation	NEAT
Training	6.69E-05	9.55E-02
Cross-validation	7.69E-01	1.08E+00
Prediction	8.21E-01	1.16E+00

TABLE V. ERRORS OBTAINED FOR SHALLOW WATER MODEL EXPERIMENT USING GRID SIZE OF 20X20

	Backpropagation	NEAT
Training	6.36E-05	3.55E-02
Cross-validation	5.05E-01	6.16E-01
Prediction	1.24E+00	1.70E+00

In relation to the number of operations that need to be performed to obtain an output, what is directly related to the neural network topology, fig. 3 summarizes the results for the evaluated grid sizes. It is clear that the neural network trained by the NEAT method was far superior, and the difference to backpropagation increases rapidly with grid size. The optimal topology presented by NEAT showed a slight growth with grid size, while the number of operations required using backpropagation, even without the hidden layer, showed a much more pronounced upward trend.

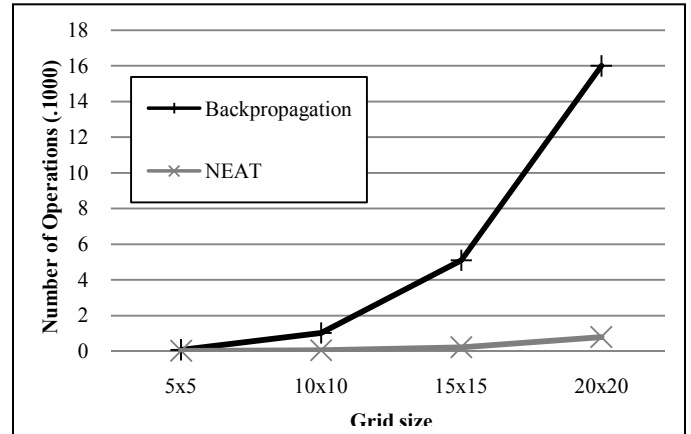


Figure 3. Processing cost variation with grid size, in terms of number of operations required.

Fig. 4 shows the experiments results considering processing time in seconds to perform the data assimilation for the whole prediction. Given the short time of the experiments, the results consisted of the average of ten times repetition of the experiment. NEAT method, even showing lower number of operations for all test cases, does not maintain the same comparative performance when processing time is considered. For grid sizes of 5x5 and 10x10, NEAT presents a slight worse performance, probably because NEAT method implementation presents a more complex structure, with several layers of abstraction, in contrast to fully connected neural network that is represented by matrices, where an update operation is solved very efficiently. But for larger states, with the increasing

advantage of NEAT method in terms of number of operations required, the time processing was better too. So even with a more complex structure, NEAT can present a better computational performance, especially in problems with larger states. The experiments were run on a computer equipped with dual Intel Xeon E5620 2.40 GHz processors and 32 GB of RAM.

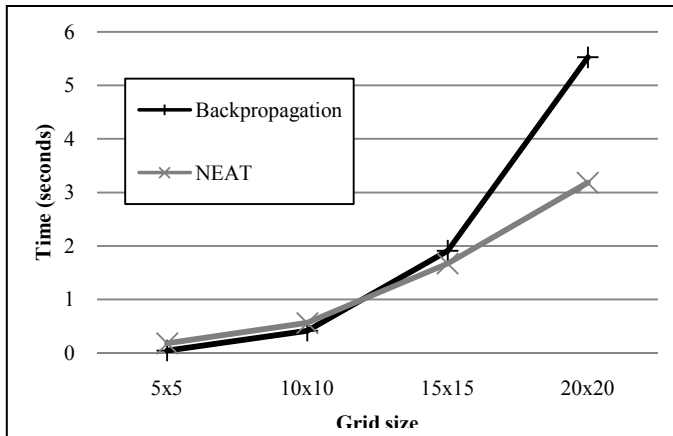


Figure 4. Processing time variation with grid size.

VI. CONCLUSION

This paper presented a new approach to perform data assimilation using the concept of minimal topology presented in NEAT method. Data for the experiments were obtained from the models of Lorenz and Shallow Water, both references in literature for testing new techniques in data assimilation. Two neural network models were applied in the experiments: neural network of fixed topology trained by backpropagation and the neuro-evolutionary method NEAT. The BLUE method was the data assimilation method to be emulated in the experiments.

Experiments were conducted to verify the effectiveness and efficiency of neural networks applied to data assimilation task. Two different models were used for testing: the Lorenz Attractor and Shallow Water model. The experiments compared the results obtained with NEAT and backpropagation neural networks, using as benchmark the BLUE. In the experiment with the Lorenz Attractor, it was found that all models of neural networks were able to emulate the data assimilation technique, but NEAT showed smaller error at lower computational cost. For the Shallow Water model, four experiments varied grid size from 25 to 400 variables. It was observed that neural network trained by backpropagation showed lower error values in all cases. For the training set, backpropagation neural network was far superior to NEAT. However, for the validation and prediction sets, the difference between the neural networks was significantly reduced. On the other hand, NEAT always got a topology with significantly fewer operations, and the computational cost difference rapidly increased with grid size. Due to its complex representation, NEAT computational performance in terms of time processing

reflected the correspondent reduction in the number of operations only for grid sizes larger than 15x15.

ACKNOWLEDGEMENTS

The authors would like to acknowledge Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq for financial support.

REFERENCES

- [1] R. Daley, Atmospheric Data Analysis, 1rd ed., vol. 1. Cambridge University Press, 1991.
- [2] A.G. Nowosad, H.F. de Campos Velho, and A. Rios Neto, "Neural Network as a New Approach for Data Assimilation", Brazilian Congress on Meteorology. Brazil, pp. 3078-3086, 2000.
- [3] R. Cintra, H. Velho, and R. Todling, "Atmospheric Data Assimilation using networks Neural: New Approach - Assimilação de Dados Atmosféricos utilizando Redes Neurais Artificiais: Nova Abordagem", Brazilian Congress of Neural Networks. Brazil, 2010.
- [4] Y. Tang, and W. Hsieh, "Applying neural network models top prediction and data analysis in meteorology and oceanography", Bulletin of the American Meteorological Society, v. 79, n. 9, p. 1855-1870, 1998.
- [5] A.G. Nowosad, A. R. Neto, and H.F. C. Velho, "Data Assimilation in Chaotic Dynamics Using Neural Networks", Third International Conference on Nonlinear Dynamics, Chaos, Control and Their Applications in Engineering Sciences, Brasil, 2000.
- [6] M. W. Gardner, and S. R. Dorling, "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences", vol. 32, Atmospheric Environment, pp. 411-420. 1997.
- [7] F. P. Harter, and H. F. C. Velho, "Different Topologies of Neural Networks for Data Assimilation", Brazilian Congress on Meteorology, 2002.
- [8] N. L. Vijaykumar, A. J. Preto, S. Stephany, A. G. Nowosad, and H. F. C. Velho, "On Optimizing a Neural Network Code for Data Assimilation", Brazilian Congress on Computing and Applied Mathematics, 2000.
- [9] F.P. Harter, and H. C. Velho, "Recurrent and Feedforward Neural Networks Trained with Cross Correlation Applied to the Data Assimilation in Chaotic Dynamic", Brazilian Journal of Meteorology, pp. 411-420. 2005.
- [10] R. S. C. Cintra, and H. F. C. Velho, "Global Data Assimilation by Artificial Neural Networks for an Atmospheric General Circulation Model: Conventional Observation", ICAI'11 - The 2011 International Conference on Artificial Intelligence, 2011.
- [11] K. Stanley, and R. Miikkulainen, "Evolving Neural Networks through Evolving Neural Networks through", The MIT Press Journals, pp. 99-127. 2002.
- [12] E. Kalnay, "Atmospheric Modeling, Data Assimilation and Predictability", 1rd ed., vol. 1. Cambridge University Press, 2002.
- [13] E. Lorenz, "Deterministic non periodic flow", Journal of the Atmospheric Physics, pp. 130-141. 1963.
- [14] J. Lu, and G. Che, "A New Chaotic Attractor Coined", International Journal of Bifurcation and Chaos, v. 12, n. 3, pp 659-661, 2002.
- [15] S. Bradford, and F. B. Sanders, "Finite-Volume Models for Unidirectional, Nonlinear, Dispersive Waves", ASCE Journal of Waterway, Port, Coastal, and Ocean Engineering, pp. 173-182. 2002.
- [16] D. B. D'Ambrosio, and K. O. Stanley, "A Novel Generative Encoding for Exploiting Neural Network Sensor and Output Geometry", Genetic and Evolutionary Computation Conference, 2007.
- [17] A. Kirillov, "AForge.NET framework", <http://www.aforgenet.com/>, Accessed 16 January 2012.
- [18] C. Rüegg, M. Cuda, and J. V. Gael, "Math.NET Numerics", <http://numerics.mathdotnet.com/>, Accessed 16 January 2012.