Proceedings of the VI Brazilian Conference on Neural Networks - VI Congresso Brasileiro de Redes Neurais
pp. 181–185, June 2–5, 2003 - Centro Universitario da FEI, Sao Paulo - SP - Brazil

Spatiotemporal Patterns Estimation Using a Multilayer Perceptron Neural Network in a Solar Physics Application

Maria Conceição de Andrade[1], Atair Rios Neto[2], Reinaldo R. Rosa[2],
Hanumant S. Sawant[1], Francisco C. R. Fernandes[1]
[1]Astrophysics Division DAS-INPE C.P. 515 12201-970 S.J. Campos, SP, Brazil
[2]Lab. for Computing and Applied Math. LAC-INPE C.P. 515 12201-970 S.J. Campos, SP, Brazil
E-mails: con@das.inpe.br, atairrn@uol.com.br, reinaldo@lac.inpe.br,
sawant@das.inpe.br, guga@das.inpe.br

Abstract

In this paper we intend to evaluate the use of multilayer perceptron neural network in a spatiotemporal patterns estimation problem and to compare the performance of the Kalman Filtering with the Backpropagation and the Levenberg-Marquardt training algorithms. The study consists of applying multilayer perceptron for estimation of the solar active regions evolution using sequential soft X-ray images obtserved by the YOHKOH solar satellite telescope. The performance test in this application is done by using the mean squared error, image visualization and the Gradient Pattern Analysis (GPA) technique, using the operator for characterization of Amplitude Asymmetric Fragmentation (AAF). The AAF operator is being used for the first time in a performance test with an Artificial Neural Network (ANN) applied in spatiotemporal patterns estimation. The results confirm the efficiency and efficacy of ANN as a tool to estimate spatiotemporal patterns in this kind of application. The tests indicate that although the Kalman Filtering showed an efficacy to learn the patterns comparable to those of the Backpropagation and the Levenberg-Marquardt algorithms, it is inefficient from the computational viewpoint in the sense that it takes a longer processing time. In addition, the ANN performance validation tests confirm the utility of the AAF operator for the performance characterization of spatiotemporal patterns algorithm generators.

1. Introduction

The implementation and performance of Artificial Neural Networks (ANN) for the solar corona spatiotemporal patterns estimation are investigated. The patterns are visualized by satellite images of the solar actives regions (http://www.solar.isas.ac.jp). To perform the patterns estimation, the Backpropagation [1], the Levenberg-Marquardt [2] and the Kalman Filtering [3] training algorithms are used and compared with each other. Based on these images, solar events forecast studies can allow us to predict the possible subsequent effects in the earth nearest regions (magnetic storms and ionospheric disturbances). The technique has been tested as a phenomenological tool to help the prediction of solar activity into the Latin-American Space Weather Forecasting Program (PLAICE) [4] and in the Brazilian Decimetric Array Project (BDA) at DAS-INPE [5][6]. Also the technique will be one of the tools in the Phenomenological Analysis System (SAF), which has been developed by Nucleus for Simulation and Analysis of Complex Systems (NUSASC) at LAC-INPE. The neural network answers will be mainly evaluated using the Amplitude Asymmetric Fragmentation (AAF) matrix computational operator [7]. This operator is applied to both images, namely to the neural network answers(estimate) and to the real images. The expectation in getting good results with ANN is based on the fact that in the last few years the ANN have been established to be an effective tool in temporal series behavior forecast. They have also been applied for some solar-terrestrial temporal series forecast (e.g. [8]). In this work we are extending this application for the spatiotemporal domain (images sequence analysis). The use of ANN in image prediction is attractive because they are capable of learning patterns, which due to their complexity are hard to analyse by other technique. They also have the ability of integrating information from samples and incorporating new characteristics without degrading the previously acquired knowledge.

2. Training Algorithms

The Backpropagation is the most common learning algorithm, however a neural network implementation based on it could take too many steps and thus a long training time. Therefore there are many studies based on heuristics or on numeric optimization technique to accelerate the algorithm convergence in order to improve its efficiency related to the time training. In order to do this, one of the chosen heuristics to this work is a definition of the term momentum. This heuristic has been demonstrated to have a good performance. We choose the Levenberg-Marquardt [9] and the Kalman Filtering [10], [11], [3] as numeric optimization techniques. Although they present a greater complexity they have been demonstrated to be efficient when compared to the gradient descent technique.

2.1 Backpropagation Algorithm

The Backpropagation is the most popular algorithm to train the multilayer neural network. It was developed by Rumelhart et al. in 1986. It implements the gradient descent technique, in order to minimize the mean square error between the desired and network output (with respect to the weights). When the network is properly trained, the Backpropagation algorithm tends to give reasonable answers when presented to new inputs that it has never seen.

2.2 Levenberg-Marquardt Algorithm

The Levenberg Marquardt is an approximation to Newton's method, an optimization technique more powerful than the descent gradient. The Levenberg-Marquardt weight actualization rule is:

$$\Delta w = (J^T\ J + \mu I)^{-1}\ J^T e\ , \tag{1}$$

where J is the Jacobian matrix of derivatives of errors with respect to the weights, μ is a scalar, and e is the error vector. If the scalar μ is large, the above expression represents the steepest descent (with step $1/\mu$), while for small μ it reduces to the Gauss-Newton method, which is more rapid and more precise, near the minimum error. Thus the goal is to transfer the learning to the Gauss-Newton method. So μ is increased or decreased after each step depending upon the case [2].

2.3 Extended Kalman Filtering Algorithm

The Kalman Filtering is an algorithm used to estimate dynamic system states from noisy measurements. Thus it is possible to use it in neural network for adjusting (estimate) weights that find a computational model for a given data set mapping such as:

$$\{(x(t), y(t)) : y(t) = f(x(t), t = 1, 2, ..., m)\}\ . \tag{2}$$

The neural network weights are the states to be estimated and the network outputs are the measurements from which the Kalman filter does the estimation:

$$\hat{y}(t) = \hat{f}(x(t), \hat{w})\ , \tag{3}$$

where $\hat{w}$ is the estimated neuron synaptic weight vector and $x(t)$ are the network inputs.

In the proposed solution [3], a linear perturbation is done in an iteration i, changing the training into the following estimation problem:

$$\bar{w} = w(i) + \bar{e} \tag{4}$$

$$\alpha(i)[y(t) - \bar{y}(t,i)] \cong \hat{f}_w(x(t), \bar{w}(i))[w(i) - \bar{w}(i)] + v(t), \tag{5}$$

where $i = 1, 2, ..., I$, $\bar{w}$ is the apriori estimate of w coming from the previous iteration, starting with $\bar{w}(1) = \bar{w}$, $\bar{y}(t,i) = \hat{f}(x(t), \bar{w}(i))$. Here $\hat{f}_w(x(t), \bar{w}(i))$ is the matrix of partial derivatives with respect to w; $0 < \alpha(i) \leq 1$ is a parameter to be adjusted in order to guarantee the hypothesis of linear perturbation, $\bar{e}$ is the apriori error distribution, and $v(t)$ are the Gaussian random variables, the error in the network output approximation.

3. Validation

3.1 Mean Square Error (MSE)

The square error E is the answer that comes from an input vector x presented to the network, producing an output signal y. Based on this error, the learning performance index is obtained using the MSE criteria:

$$E_{med} = \frac{1}{N}\left[\sum_{n=1}^{N} E(n)\right]. \tag{6}$$

This is the most commonly used technique to validate the neural network results. However the MSE value by itself is not sufficient to represent the visual quality of an image, i.e. the MSE is not always capable of determining the necessary precision to validate the neural network answers [12]. Therefore, to complete the neural network answers validation, a new technique is applied in addition to the estimated images visualization. This new technique is explained next.

3.2 Gradient Pattern Analysis and Amplitude Asymmetric Fragmentation Operator

The Gradient Pattern Analysis (GPA) was developed in order to characterize complex regimes in the spatiotemporal domain [13],[7],[14]. It was first applied to the identification of weak turbulence patterns in the solar coronal plasma [15]. Nowadays this technique is being applied not only to the characterization of non linear phenomena observed in the spatiotemporal domain (intermittence, spatiotemporal chaos) but also to the characterization of complex structures (convex asymmetry, labyrinth and non-homogeneous roughness) in a strictly spatial domain, such as the study of porosity in semiconductors [16],[17]. This technique involves the application of two computational operators which act over the gradient field of a given matrix. One of these operators is the AAF which computes the broken symmetry in the gradient field of a matrix. This operator working over the spatiotemporal domain is efficient to detect small non-linear fluctuations which are not visualized with accuracy during the spatiotemporal patterns evolution. In this work the AAF operator was used to test, for the first time, the neural network estimation performance working in the spatio temporal domain.

The AAF operator is capable of locally quantifying the symmetry breaking in the image field gradient. It turns a matrix M into a triangulation field with L points and I lines, where the L points correspond to the asymmetric vector numbers of the matrix gradient field and the

I lines correspond to the number of triangulation lines between the L points. The measure of the asymmetric fragmentation degree is given by the value $F_A = (I - L)/L$. This parameter shows high sensitivity for characterization of small alterations in a matrix gradient field, reflecting the possible symmetry breaking that can occur in the spatiotemporal domain [7].

4. Pattern set and training methodology

The patterns used in this work are a set of soft X-ray solar corona images obtained from the YOHKOH satellite (http://www.solar.isas.ac.jp). An image series with a little pattern alteration was used. These images observed in 04/21/92 from 11h00′ to 13h41′ UT, are available in the ASCII format, as 128×128 lattice (16384 pixels). The image series has a different temporal resolution, since the images are not taken at regular time intervals. Therefore it was necessary to interpolate the images in order to make the intervals equal, for making the training and test set. A linear interpolation with a time interval of 3 minutes between the images was used.

Two kinds of approaches were used for training. In the first one, due of computer memory limitation, the images were resized into 64×64 pixels. The training data set is formed by 5 neural network patterns, each one made up of a pair of input and output images, where the output image of the first pattern is the input of the second one, and so on until the last pattern. Each pixel of the input pattern is seen as a node, completing 4096 input nodes mapped to 4096 output nodes.

The second approach is constructed in the spatial domain. Starting from the first pixel of the images, given a temporal series to be analysed, one defines as a neural network input pattern the pixels in the same spatial position in the image of the temporal series, and the output is the pixel of the last image in the sequence. Once the images are 128×128 pixels in a series of five images, 16384 neural network patterns are formed with four inputs and one output each one. Figure 1 shows this kind of architecture.

5. Results

5.1 First approach

The learning algorithm with better training performance to this approach was Backpropagation with momentum. The best training, with an architecture 64-15-64 nodes in each layer, took about 15000 epochs in 25 hours approximately. The ANN parameters were: learning rate 0.09, the momentum constant 0.5, the threshold 10^{-6} and the activation functions sigmoid and the identity were used on the hidden and the output layer respectively. The values of MSE were 9.75×10^{-5} and 1.2×10^{-3} (training and test respectively). The values of F_A were 1.9813 to the real and 1.9799 to the estimated image. Figure 2 shows the temporal sequence images (64×64 pixels) used as input-output pattern in all training

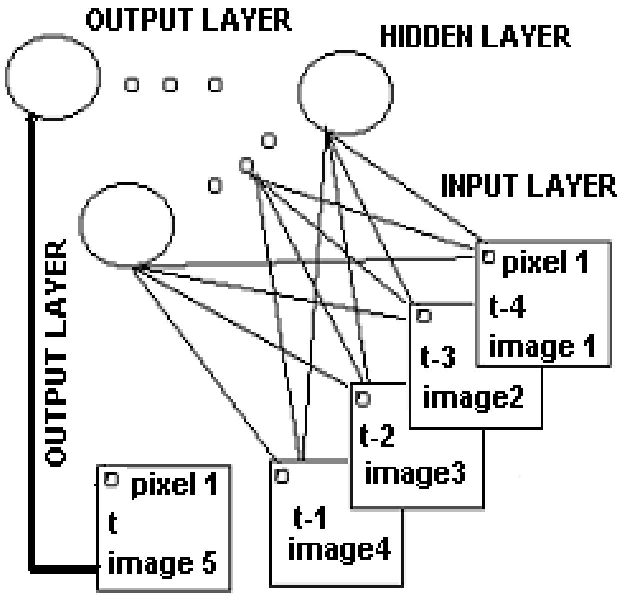

Figure 1: Example of an ANN architecture formed by the second approach

during this approach. Figure 3 shows the image used for the test, the image that the neural network should estimate and the neural network answer. The simulations were made in the microcomputer (architecture CISC compatible with IBM-PC standard) Pentium III 1.1GHz and 512 Mb of RAM memory .The Levenberg-Marquardt and the Kalman Filtering algorithms did not show a good performance to this neural network architecture since they use a lot of memory in the weight correction computation.

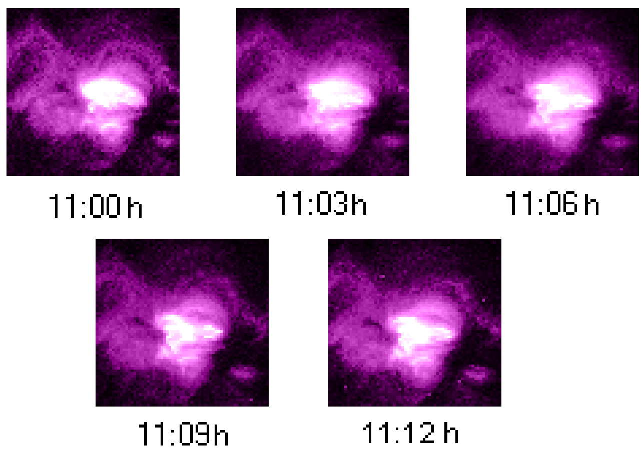

Figure 2: ANN training patterns used for training with the first approach

5.2 Second Approach and Kalman Filtering Algorithm

In the second approach the neural network architecture was defined as 18 neurons in the hidden layer, 4

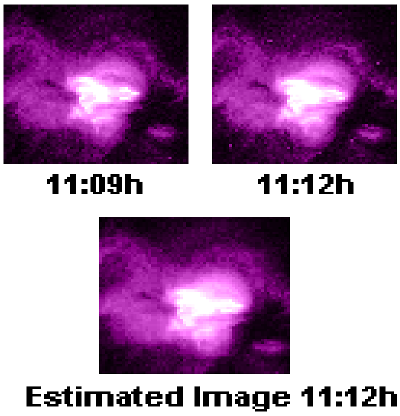

Figure 3: Training results with the first approach: image used in the test (11:09h), image to be estimated (11:12h) and estimated image with Backpropagation algorithm (11:12h below)

nodes in the input layer, one node in the output layer and 16384 patterns. The activation functions sigmoid and the identity were used on the hidden and the output layer respectively. The best training of the Kalman Filtering algorithm with the architecture defined above was made with a learning rate 0.1, noise value 0.1 and threshold 10^{-6}. The training took 20 epochs in 30 minutes approximately, with the training MSE of 2.1×10^{-3} and the test MSE of 1.9×10^{-5}. The values of F_A are 1.99202 (for the real image) and 1.99183 (for the estimated image). Figure 4 shows the temporal sequence images (128×128 pixels) used as input-output patterns in the second approach training, and figure 5 shows the temporal series patterns used for the test (13:27h to 13:36h), the image that the neural network should estimate (13:39h) and the neural network answer with second approach and Kalman Filtering algorithm.

5.3 Second Approach and Levenberg-Marquardt Algorithm

The architecture of the neural network and the activation functions for this training were the same as used in the Kalman Filtering. The best training of the Levenberg-Marquardt algorithm was made with learning rate 0.1 and threshold 10^{-6}. The training took 15 epochs in 10 minutes approximately, with the training MSE of 1.89×10^{-3} and the test MSE of 1.2×10^{-3}. The values of F_A are 1.99202 (for the real image) and 1.99315 (for the estimated image). Figure 3 shows the neural network answer with second approach and Levenberg-Marquardt algorithm.

5.4 Second Approach and Backpropagation Algorithm

The neural network architecture and the activation functions for this training were the same as used in both the above cases. The best training of the Backpropagation algorithm was made with learning rate 0.5, momentum constant 0.9 and threshold 10^{-6}. The training took about 100 epochs in 10 minutes approximately, with the training MSE of 4.91×10^{-3} and the test MSE of 4.00×10^{-3}. The values of F_A are 1.99202 (for the real image) and 1.99396 (for the estimated image). Figure 3 shows the neural network answer with second approach and Backpropagation algorithm.

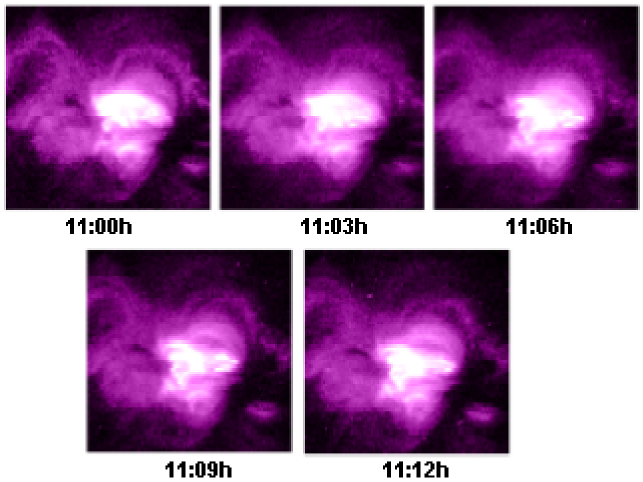

Figure 4: Training patterns with the second approach

6. Concluding Remarks

We presented in this paper an evaluation in using multilayer perceptron neural network in a spatiotemporal patterns estimation problem. To do this we considered the use of the learning algorithm Kalman Filtering comparing its results with Levenberg-Marquardt and Backpropagation with momentum. To validate the ANN performance training, the Gradient Pattern Analysis technique was used for the first time, proving to be a powerful technique for the performance characterization of algorithm generators of spatiotemporal patterns.

With the first approach, throughout the training period, it was noticed that although Kalman Filtering and Levenberg-Marquardt training algorithms are robust they turned out to be inefficient from the computational memory viewpoint. The Backpropagation with momentum showed the best performance and provided a good answer.

The computatitonal memory problem was solved using the second approach. The analysis of the second approach results showed that the Kalman Filtering presented the best answer (MSE, F_A and image visualization) and an efficacy to learn the patterns comparable to those of the Levenberg-Marquardt and of the Backpropagation algorithms. However the Kalman Filtering still must be optimized in its processing time to improve its efficiency.

From the preliminary results one can conclude that the artificial neural network methodology is promising

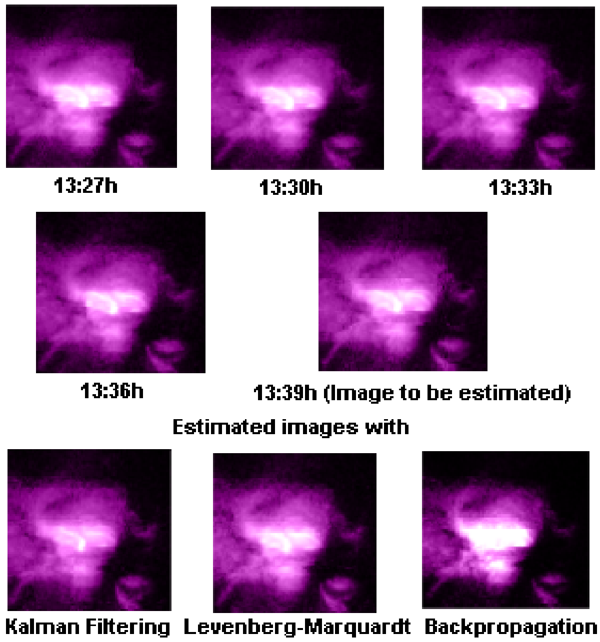

Figure 5: Training with the second approach: ANN test patterns (13:27h-13:36h); the image to be estimated for the ANN (13:39h), the estimated images with Kalman Filtering, Levenberg-Marquardt and Backpropagation algorithms respectively

for the solar application. The three algorithms have been shown to be effetive tools to train neural network for short-term forecast (three minutes) for spatiotemporal patterns produced by non-linear systems. However their performance should be specifically evaluated for other phenomenological forecast application.

Acknowledgements

The authors are grateful for the support received from the Astrophysics Division and Lab. for Computing and Applied Math. of INPE in order to develop this work. RRR is grateful for FINEP and FAPESP support.

References

[1] D. E. Rumelhart and G. E. Hinton and R. J. Willims. Learning Representations of Backpropagation Errors. *Nature* **323**:533-536, 1986.

[2] M. T. Hagan and M. B. Menhaj. Training Feedforward Networks with The Marquardt Algorithm. *IEEE Transactions on Neural Networks* **5**(6), 1994.

[3] A. Rios Neto. Stochastic optimal linear parameter estimation and neural nets training in systems modeling. *Journal of the Braz. Soc. Mechan. Sciences*, **19**, (2): 138-146, 1997.

[4] R. R. Rosa and H. S. Sawant and J. R. Cecatto and C. Rodrigues Neto and V. A. Lopes and K. R. Subramanian and F. R. Fernandes and J. H. Saito and C. E. Moron and M. L. Mucheroni and N. Furuya and N. Mascarenhas. Phenomenological Dynamics of Coronal Loops Using a Neural Network Approach. *Adv. Space Res.*, **25**(9):1917-1921, 2000.

[5] H. S. Sawant and E. Ludke and K. R. Subramanian and F. R. Fernandes and J. R. Cecatto and R. Rosa. A high resolution decimetric solar radioheliograph. *Astronomical Soc. of the Pacific Conference Series*, bf206:341-346, Edited by R. Ramaty and N. Mandzhavidze, 2000a.

[6] H. S. Sawant and K. R. Subramanian and E. Ludke and J. H. Sobral and G. Swarup and F. C. Fernandes and R. Rosa and J. R. Cecatto. Brazilian Decimetric Array. *Advances in Space Research*, **25** (9):1809-1812, 2000b

[7] R. Rosa and A. S. Sharma and J. A. Valdivia. Characterization of Asymetric Fragmentation Patterns in Spatially Extended Systems. *Intern. Journal of Modern Physics C* **10**(1):147-163, 1999.

[8] K. P. Macpherson and A. J. Conway and J. C. Brown. Prediction of Solar and Geomagnetic Activity Data Using Neural Networks. *Journal of Geophysical Research* **100**(A11):21735-21744, 1995.

[9] D. Marquardt. An algorithm for least squares estimation of non-linear parameters. *J. Soc. Ind. Appl. Math.* 431-441, 1963.

[10] S. Singhal and L. Wu. Training feedforward networks with the extended Kalman filter. *Proceedings of IEEE Internat. Conference on Acoustics Speech and Signal Proc.* 1187-1190, Glasgow, Scotland: IEEE Press, 1989.

[11] R. S. Scalero and N. Tepedelenlioglu. A fast new algorithm for training feedforward neural network. *IEEE Transactions on Signal Proc.* **40**(1):202-210, 1992.

[12] F. Russo. Hybrid neuro-fuzzy filter for impulse noise removal. *Pattern Recognition* **32**:1843-1855, 1999.

[13] R. Rosa and C. Rodrigues Neto and F. Ramos and A. S. Sharma and J. A. Valdivia. Computational operators for dynamical complex pattern recognition. In EPS-Modelling Collective Phenom. in Complex Systems, v. 22F, Pub. by European Phys. Soc., p. 304-305, 1998b.

[14] F. M. Ramos and R. R. Rosa and C. Rodrigues Neto and A. Zanandrea. Generalized complex entropic form for gradient pattern analysis of spatio-temporal dynamics. Physica A, v. 283. p. 171-174, 2000.

[15] R. Rosa, A. S. Sharma, J. A. Valdivia. Characterization of localized turbulence in plasma extended systems. *Physica A* **257**(1-4):509-514, 1998a.

[16] A. Ferreira da Silva and R. Rosa and L.S. Roman and E. Veje. Characterization of asymmetric fragmentation patterns in SFM images of porous silicon. *Solid State Comm.* **113**(12):703-708, 2000a.

[17] A. Ferreira da Silva and R. Rosa and P. W. Machado and F.M. Ramos and C. Rodrigues Neto and L.S. Roman and E. Veje. Asymmetric nonlinear amplitude in patterns of porous silicon. *Physica A* **283**:223-227, 2000b.