Journal of Computer Science

Original Research Paper

A New Methodology for Neural Network Training Ensures Error Reduction in Time Series Forecasting

Paola A. Sánchez-Sánchez and José Rafael García-González

Facultad de Ingeniería, Universidad Simón Bolívar, Barranquilla, Colombia

Article history
Received: 27-04-2017
Revised: 22-06-2017
Accepted: 30-06-2017

Corresponding Author:
Paola A. Sánchez-Sánchez
Facultad de Ingeniería,
Universidad Simón Bolívar,
Barranquilla, Colombia
Email:
psanchez9@unisimonbolivar.edu.co

Abstract: Artificial Neural Networks (ANN) consists of some components, such as architecture and learning algorithm. These components have a significant effect on the performance of the ANN, but finding good parameters is a difficult task to achieve. An important requirement for this task is to ensure the reduction of error when inputs and/or hidden neurons are added. In practice, it is assumed that this requirement is always true, but usually it is false. In this paper, we propose a new algorithm that ensures error decrease when input variables and/or hidden neurons are added to the neural network. The behavior of two traditional algorithms and the proposed algorithm in the forecast of Airline time series were compared. The empirical results indicate that the proposed algorithm allows a steady decrease of fit error in all cases, where de most important and differentiable feature is the fact that reach values close to zero, which is not true for the other algorithms. Therefore, it can be used as a suitable alternative algorithm, especially when it needs a good fit.

Keywords: Artificial Neural Networks (ANN), Time Series Forecasting, Learning Algorithms, Error Reduction

Introduction

Time series prediction with neural networks has been an accepted practice in the literature by its capabilities of generalization and fitting. However, alarge number of factors must be determined in the process of building neural network model which often leads to inconsistent results (Zhang *et al*., 1998). Fitting capacity of a neural network is affected by the configuration used, especially in relation of hidden neurons and input variables numbers (Crone and Korentzes, 2009; Murata *et al*., 1994); when the number of parameters model increases, it favors the network learning and therefore the fitting is best. In theory, an iterative process of adding parameters (inputs and hidden neurons) should lead to systematic reductions in the fit error. An appropriate process for estimating the parameters of a neural network is the starting point for determining the model. Hornik *et al*. (1989), have shown that artificial neural networks multilayer perceptron are universal approximators of functions, but it is not known how one may estimate the parameters of the neural network in such a way that error decreases. It seems always possible for the scientific community to obtain a model that reaches the desired degree of accuracy.

It is known that one of the reasons for the poor performance of a neural network is related to the strengths and weaknesses of the training algorithm used, i.e., their ability to avoid local minima in fitting function and ease of calibration parameters. This has motivated the approach of many new algorithms; they do a search of the optimum computationally faster and allow finding optimal quality better than other methods. However, these algorithms do not take into account the practical implications of those networks are universal approximators of functions and error reduction when parameters are increased. This article presents a novel algorithm that allows the systematic reduction of error as they are added to the neural network parameters.

The originality, significance and relevance of this work is based on the following:

- It evaluates the theoretical modeling of neural networks, such as the error reduction in addictive process parameters
- It develops an algorithm to fulfill the expected theoretical behavior

Problematic and Proposed Methodology

Architecture of the Multilayer Perceptron - MLP

This type of neural network architecture consists of:

- An input layer, consisting of the P lags of the time serie y_t
- A hidden layer with H neurons. The transfer function $g(\cdot)$ used in the connections β_h is logistics
- An output layer with one neuron

The present value of a time series is y_t a nonlinear function of its past values $y_{t-1},\ldots, y_{t-p}$, which is defined as Equation 1:

$$y_t = \sum_{h=1}^{H}\left\{\beta_h \times g\left[\omega_h + \sum_{p=1}^{P}\alpha_{p,h}y_{t-p}\right]\right\} + e_t \quad (1)$$

where, e_t represent the errors (independent and identically distributed random variables - i.i.d).

The optimization of the model parameters depends on the error between the desired and predicted by the network during training and is associated with the configuration of the neural network; the selection of the optimal number of variables of the model is directly related to the training process used. As discussed Qi and Zhang (2001), there is a close relationship between model performance and the selection of the values of P and H.

Estimation of the Parameters of the MLP as an Optimization Problem

With the design of an artificial neural network which is to be achieved for certain values lagged of the variable explained, it is capable of approximating the current value of the time series with a desired accuracy (Zhang *et al*., 1998). As has been established by Hornik *et al*. (1989): *"…standard multilayer feed forward networks with as few as one hidden layer using arbitrary squashing functions are capable of approximating any Borel measurable function from one finite dimensional space to another to any desired degree of accuracy, provided many sufficiently hidden units are available…*", implies that a constructive process of adding hidden neurons(variables) will allow a sequential reduction of the fit error of the model of the series to a desired level, in other words, a model with more hidden neurons should have better fit of the training data than one with less hidden neurons. However, the study does not indicate how to estimate the parameters of the neural network.

Following the guides indicated by Hornik *et al*. (1989), the error of fitting the model must decrease, or at least stay the same, when adding hidden neurons or input. The reasoning is as follows:

- Be a MLP with M hidden neurons and the first P-series lags MLP(P,H)
- The parameters (P,H) were estimated minimized any error function. The error is denoted by E(P,H)
- Be a MLP with an additional hidden neuron, i.e., MLP(P,H +1). It holds that E(P,H) >= E(P,H +1),by contradiction: Suppose that E(P,H) < E(P,H +1), the MLP(P,H +1) model can be obtained MLP(P,H) by adding a hidden neuron, to retain the weights of parameters of MLP(P,H) model and the connection H +1 becomes zero, therefore has to be E(P,H) = E(P,H +1) and consequently, the error must remain the same or be reduced by optimizing the MLP(P,H+1) model
- Be a MLP with one additional entry, i.e., MLP(P +1,H). It holds that E(P,H) > = E(P +1,H)

This implies that a process of adding hidden neurons will allow (at least theoretically) a sequential reduction of the error of fitting the model to a level arbitrarily close to zero, in other words, a model with more hidden neurons should better fit the training data to a model with fewer hidden neurons.

In literature, statistical error reduction adjustment to increase the complexity of the model is a well-known concept. The MLP(P,H) model is called restricted model, sub model or nested model regarding MLP(P,H +1) and MLP(P +1,H) models, which are known as complete models.

In conclusion, for a multilayer perceptron and under a constructive scheme of hidden addiction and inputs, there should be a systematic reduction of the error of adjustment every time you add parameters to a level of accuracy desired setting (Sánchez and Velásquez, 2010).

Proposed Algorithm

From the above discussion, there are two requirements:

- That error decreases as hidden neurons are added
- That error decreases as entries are added

Suppose we want to approximate the behavior of a time series by a neural network MLP. The training of the network can be formulated as an unconstrained optimization problem:

$$\min_{P,H,w}\sum_{i=1}^{P}(y_{i\ calc} - y_i)^2 \quad (2)$$

Subject to:

$$y_{i\ calc} = f\left(wx^i\right), w \in \mathcal{R}^{PxH}, y_{i\ calc} \in \mathcal{R}^P, x^i \in \mathcal{R}^H \quad (3)$$

In practice, it requires an iterative procedure to solve Equation 2-3, where the weights w, the input number of hidden neurons *P* and *H* are calculated in order to minimize the fitting. It is assumed as a starting point a network with minimal topology and the optimal weights are calculated in such a model, subsequently added to the model parameters (increase inputs or hidden neurons) and weights are optimized resulting in a new model, this process is repeated until an acceptable solution is found.

Methodology for Error Reduction when Adding Hidden Neurons

A way to ensure that error decreases as neurons are added in the hidden layer originates from the comparison of restricted and full models. The equivalence of a restricted model and full model occurs when the weights of the new connections of the full model are zero, i.e., E(*P*, *H*) = E(*P*, *H* +1) [freeze weights].

After keeping the weights, the full model is optimized, ensuring that the error of this model will be less, or at least equal to, the restricted model error.

As shown before, the process of "freezing of weights" (including the subsequent optimization) can be seen as a smart choice of the initial weights of the full model, given that the weights are the best initial values that can be obtained and also ensure, an adjustment error at least equal to, the constrained model.

Algorithmically the process of adding hidden neurons above can be represented as Algorithm 1.

The algorithm of neural network configuration with a fixed number of inputs "M" and a hidden neuron defines a maximum number of neurons to add "N" and one output. The model is optimized using a standard algorithm for training neural networks. Subsequently, a hidden neuron is added to the previous model.

The weights of the above connections are frozen and new connections, generated by the additional input of the neuron, become zero.

Algorithm1: The error reduction algorithm when added in the hidd en layer neurons

```
1: Mod_Max ← Mod(M,N,1)
2: P ← M, Q ← 1, Mod_last ← [ ]
3: Mod_Actual ← Optim (Mod(P,Q,1))
4: Do While Mod_Actual =! Mod_Max or
   (Mod_last.error – Mod_Actual.error) < Tol
5:    Mod_last ← Mod_Actual
6:    Mod_Actual ← Add_neuron(Mod_Actual, P, Q+1)
7:    Optim(Mod_Actual)
8: Loop
```

The resulting model is optimized. It continues to add neurons and optimizing the new model until you reach the number of neurons initially defined or until the gain in the model error by adding a neuron is less than a specified tolerance.

Methodology for Error Reduction by Adding Entries

The optimization of the set of input variables is a more complex task than that of hidden neurons, since this does not necessarily imply that the entries are contiguous. Furthermore, when combined with the optimization process of hidden neurons, the size of the search space has an incremental complexity.

A constructive process of adding inputs and hidden neurons would test and compare in each case all possible models, which is not an efficient process.

To ensure that error decreases as input variables are added, it is necessary in order to prove that among all possible combinations of models, the selected model has the lowest error. As for the addition of hidden neurons, the addition of entries, a preservation process optimized weights of the network parameters to zero and maintaining the remaining connections ensure that increasing the configuration complete model has an error not as good or equal to the restricted model (model with less input). This is true to the extent that such tickets are spiked sequentially.

Such changes to the Algorithm1 the following points lie in [correspond to lines of code 2, 8-13 in Algorithm 2]:

- The number of inputs and neurons in the model is variable, it is part of a configuration of an input and hidden neuron and has the highest levels of "M" and "N", respectively
- For each neuron, inputs are added 1 to reach the peak "M". In each new configuration, generated weights are kept and the old settings and zeros are those of the new connections
- The process stops when it reaches the number of inputs and neurons initially defined

Algorithm 2: The error reduction algorithm when adding neurons in the hidden layer and input variables

```
1: Mod_Max ← Mod(M,N,1)
2: P ← 1,Q ← 1, Mod_last ← [ ]
3: Mod_Actual ← Optim (Mod(P,Q,1))
4: Q ← 0
5: Do While Mod_Actual =! Mod_Max or
   (Mod_last.error –Mod_Actual.error) < Tol
6:    Mod_last ←Mod_Actual
7:    Mod_Actual ← Add_neuron(Mod_Actual, P, Q+1)
8:    Mod_Temp ← Mod_Actual
9:    for P in 1:M do
10:      Mod_ Temp ← Add_input(Mod_ Temp, P, Q)
11:      Optim(Mod_ Temp)
12:   end for
13:   P ← 1
14: Loop
```

Experimentation and Results

This experiment evaluates if they are added as neurons in the hidden layer and inputs to the network, there is a reduction of the adjustment error of the model to the time series. For the log airline series, researched by Faraway and Chatfield (1998) and Nam and Schaefer (1995), conducted an experiment:

- It considered as inputs models with lags ranging from 1 to 13
- It considered models lag from one to eight neurons in the hidden layer
- Each model was optimized separately, with the generalized delta rule algorithms, Rprop and the proposed algorithm
- Conducted a procedure where each entry is added sequentially hidden neurons to the limit. For each configuration, the best model is the one with the MSE

To evaluate the performance of the algorithms the 130 possible models were built and optimized that make up the space of models.

Figure 1 shows the behavior of the proposed algorithm in terms of parameters addition. Since Fig. 1 is extracted the follows:

- The adjustment error is reduced as hidden neurons are added
- The adjustment error is reduced as lags are added (entries)
- Zero errors are achieved found

Table 1. Values of comparison of proposed algorithm with delta rule and Rprop for airline series

	Delta rule			Rprop			Proposed algorithm		
Inputs	H1	H5	H8	H1	H5	H8	H1	H5	H8
Lag1	0,0677	0,0107	0,0106	0,0156	0,0106	0,0105	0,001196	0,001196	0,001196
Lag1-2	0,0835	0,0112	0,0111	0,0160	0,0096	0,0095	0,001140	0,001058	0,001009
Lag1-3	0,0718	0,0429	0,0189	0,0154	0,0096	0,0093	0,001123	0,000848	0,000801
Lag1-4	0,0781	0,0725	0,0621	0,0167	0,0094	0,0085	0,001114	0,000540	0,000461
Lag1-5	0,0864	0,0766	0,0817	0,0140	0,0087	0,0087	0,000995	0,000309	0,000239
Lag1-6	0,0899	0,0757	0,0649	0,0165	0,0087	0,0084	0,000988	0,000253	0,000207
Lag1-7	0,0900	0,0845	0,0681	0,0130	0,0094	0,0085	0,000981	0,000160	0,000147
Lag1-8	0,0991	0,0964	0,0823	0,0167	0,0085	0,0079	0,000946	0,000120	0,000094
Lag1-9	0,1037	0,0872	0,0986	0,0117	0,0073	0,0065	0,000664	0,000096	0,000087
Lag1-10	0,0918	0,0803	0,0591	0,0121	0,0075	0,0058	0,000630	0,000073	0,000065
Lag1-11	0,0922	0,0754	0,0673	0,0106	0,0053	0,0043	0,000365	0,000064	0,000060
Lag1-12	0,0833	0,0661	0,0498	0,0109	0,0038	0,0032	0,000267	0,000059	0,000043
Lag1-13	0,0849	0,0736	0,0682	0,0071	0,0040	0,0035	0,000159	0,000054	0,000023

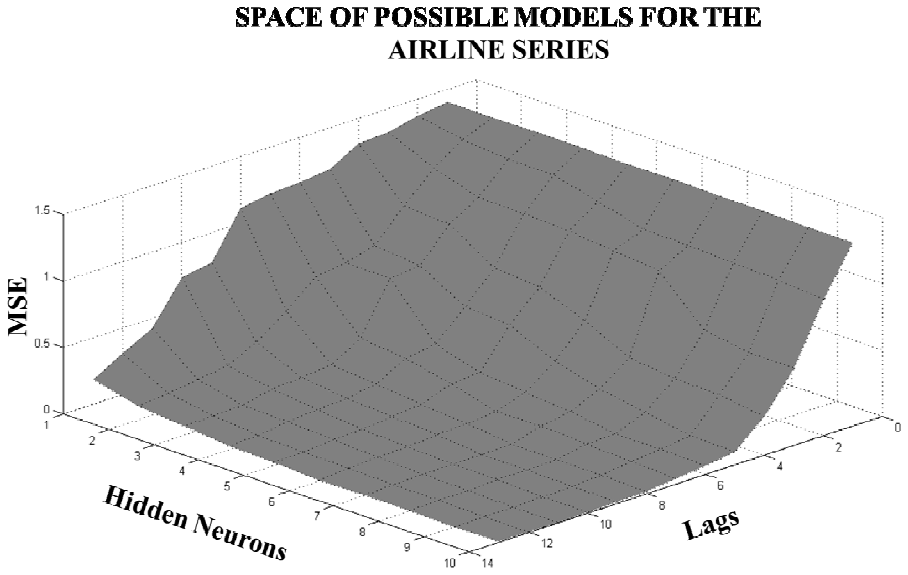

Fig. 1. Space of possible models for the Airline series

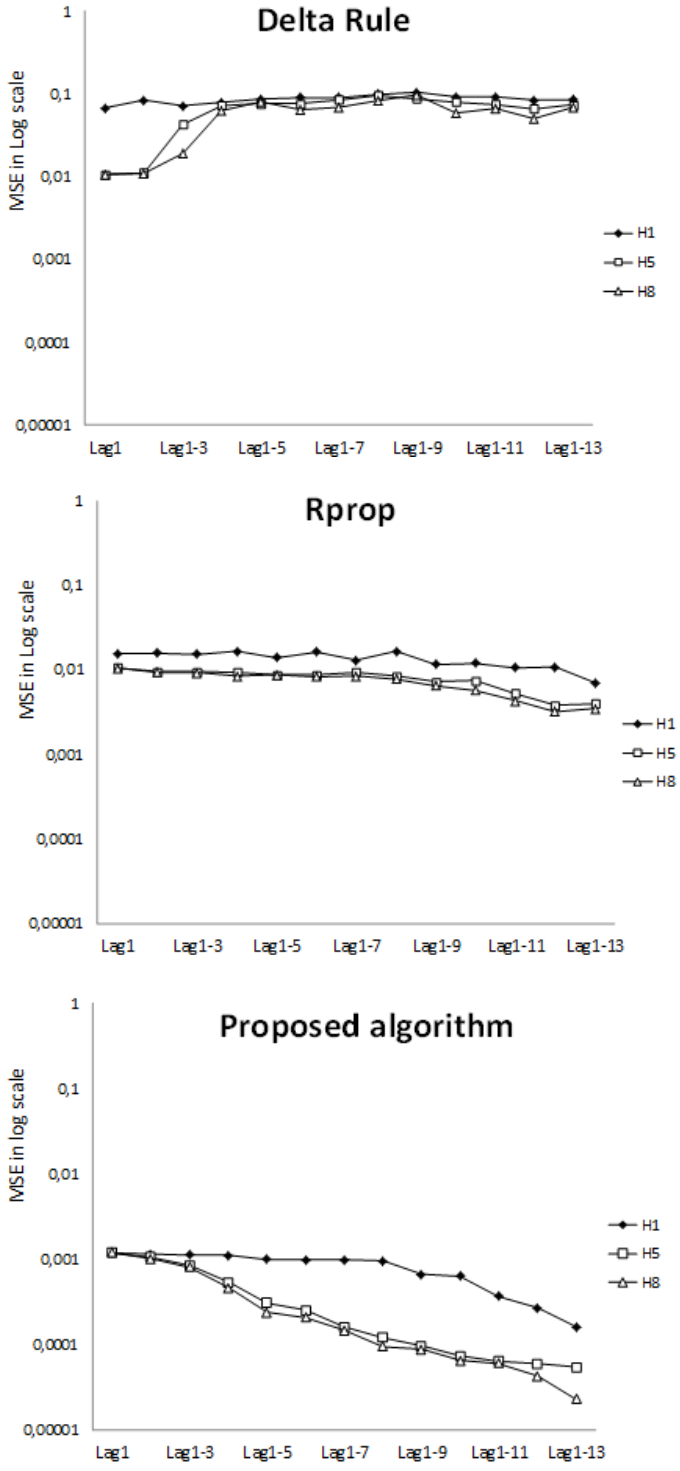

Fig. 2. Comparison of proposed algorithm with Delta Rule and Rprop for Airline Series

Figure 2 and Table 1, contrasts the adjustment error values achieved, for configurations of 1, 5 and 8 neurons and different number of lags for the Delta Rule, Rprop and proposed algorithms (Igel and Husken, 2003; Anastasiadis *et al*., 2003):

- The Delta Rule algorithm exhibits anomalous behavior by presenting higher error rates as the set of entries grows, also from the addition of the third lag of the error behavior is variable (growth and decreases)

- For the Rprop algorithm, no significant difference between using one five lags, such that different configurations of relevant entries and hidden neurons lead to similar error rates
- Delta Rule and Rprop algorithms do not show the theoretical behavior, because adding parameters have not reduced the error
- Models generated by the algorithm are competitively best in all cases, those obtained by the Delta Rule and Rprop algorithms, which is reflected by the marked decrease in the levels of the fit error

The most important and differentiable feature to rescue the implementation of the proposed algorithm is the fact that they reach values close to zero, which is not true for other algorithms.

Conclusion

This paper developed an algorithm that allows the systematic reduction in fit error when parameters are added to the neural network. The proposed algorithm differs from traditional architectures by including a constructive strategy, with two requirements that error decrease as input variables are added and when hidden neurons are added. Both requirements are achieved by including the conservation of weights for the inputs and hidden neurons; That is, for each hidden neuron in the new model, the weights of the parameters of the previous model (model with fewer inputs) are conserved and the weights of the new connections are zeroed, prior to the optimization of the model. When neurons are added, the preservation with respect to the previous model is continued with an input. The optimization of each new model, part of the error value reached by the previous model and, therefore, must decrease or remain the same.

The experimental application of algorithm proposed to the Airline time series presents very satisfactory results, since there are always gains when adding parameters, when compared to traditional algorithms as delta rule and Rprop, systematically reducing fit error. In addition, it is possible to reach levels near zero in error, which shows that it is possible to reach desired levels of fit error, even when the goal of error is zero; this is not achieved with other algorithms. This ratifies, that the proposed algorithm fulfills the criterion of universal approximation of functions.

The originality and importance of this contribution lies in the following:

- Other techniques that preserve the value of weights, in the proposed procedure, the parameter values are used as starting points for the weights of the new model prior to optimization, being an intelligent choice, given that such parameters are the best initial values that can be obtained
- A suitable set of initial values, coupled with an optimization process of fit error, ensures the success of the optimization strategy in the current model, since it is part of an optimal value of error in question and it must decrease, or at least remain the same when adding hidden neurons
- It is considered a constructive set of hidden neurons and inputs for the specification of neural network models that achieves the desired level of accuracy, to continue adding parameters
- Highly satisfactory experimental results are obtained by always achieving reductions in the fit error and reaching accuracy levels close to zero in the error for the Airline time series

Finally, the proposed algorithm presents a strength in the fit error, obviating the presence of the over fitting problem; as future work arises, i. Assess whether the problem arises, ii. Measure their impact, iii. Propose ways of addressing it and iv. Include the solutions in the algorithm.

Acknowledgment

Authors would like to acknowledge the Vicerrectoría de Investigación e Innovación, Universidad Simón Bolívar, Barranquilla - Colombia for providing the support and tools for the development of this research.

Author's Contributions

All authors made substantial contributions to this paper.

Paola A. Sánchez-Sánchez: Contributed to research design, literature review, implementing the proposed algorithm, analyzing the results, compare with other existing algorithms and writing of the manuscript.

José Rafael García-González: Contributed to conceptualization, co-implementing the proposed algorithm and analyzing the results, writing of the manuscript, editing and reviewing.

Ethics

The authors confirm that this manuscript has not been published elsewhere and that no ethical issues are involved.

References

Anastasiadis, A., G. Magoulas and M. Vrahatis, 2003. An efficient improvement of the Rprop algorithm. Proceedings of the 1st International Workshop on Artificial Neural Networks in Pattern Recognition, Sept. 12-13, At Florence, Italy, pp: 197-201.

Crone, S. and N. Korentzes, 2009. Input-variable specification for neural networks-an analysis of forecasting low and high time series frequency. Proceedings of the International Joint Conference on Neural Networks, Jun. 14-19, IEEE Xplore Press, pp: 3221-3228. DOI: 10.1109/IJCNN.2009.5179046

Faraway, J. and C. Chatfield, 1998. Time series forecasting with neural networks: A comparative study using the airline data. Applied Stat., 47: 231-250. DOI: 10.1111/1467-9876.00109

Hornik, K., M. Stinchcombe and H. White, 1989. Multilayer feed forward networks are universal approximators. Neural Netw., 2: 359-366. DOI: 10.1016/0893-6080(89)90020-8

Igel, C. and M. Husken, 2003. Empirical evaluation of the improved Rprop learning algorithms. Neurocomputing, 50: 105-123. DOI: 10.1016/S0925-2312(01)00700-7

Murata, N., S. Yoshizawa and S. Amari, 1994. Network information criterion determining the number of hidden units for an artificial neural network model. IEEE Trans. Neural Netw., 5: 865-872. DOI: 10.1109/72.329683

Nam, K. and T. Schaefer, 1995. Forecasting international airline passenger traffic using neural networks. Logist. Transport. Rev., 31: 239-251.

Qi, M. and P. Zhang, 2001. An investigation of model selection criteria for neural network time series forecasting. Eur. J. Operat. Res., 132: 666-680. DOI: 10.1016/S0377-2217(00)00171-5

Sánchez, P. and J.D. Velásquez, 2010. Problemas de investigación en la predicción de series de tiempo con redes neuronales artificiales. Rev. Avances Sistemas Inform., 7: 67-73.

Zhang, G., B. Patuwo and M. Hu, 1998. Forecasting with artificial neural networks: The state of the art. Int. J. Forecast., 14: 35-62. DOI: 10.1016/S0169-2070(97)00044-7