Available online at www.sciencedirect.com

SCIENCE DIRECT®

Advances in Space Research 36 (2005) 2480–2485

ADVANCES IN SPACE RESEARCH (a COSPAR publication)

www.elsevier.com/locate/asr

Neural network development for the forecasting of upper atmosphere parameter distributions

Jeffrey D. Martin *, Yu T. Morton, Qihou Zhou

Department of Electrical and Computer Engineering, Miami University, 119 Kreger, Oxford, OH 45056, USA

Received 19 October 2002; received in revised form 26 December 2003; accepted 27 December 2003

Abstract

This paper presents a neural network modeling approach to forecast electron concentration distributions in the 150–600 km altitude range above Arecibo, Puerto Rico. The neural network was trained using incoherent scatter radar data collected at the Arecibo Observatory during the past two decades, as well as the Kp geomagnetic index provided by the National Space Science Data Center. The data set covered nearly two solar cycles, allowing the neural network to model daily, seasonal, and solar cycle variations of upper atmospheric parameter distributions. Two types of neural network architectures, feedforward and Elman recurrent, are used in this study. Topics discussed include the network design, training strategy, data analysis, as well as preliminary testing results of the networks on electron concentration distributions.
© 2005 COSPAR. Published by Elsevier Ltd. All rights reserved.

Keywords: Space weather; Neural networks; Upper atmosphere; Electron concentration

1. Introduction

This study focused on the application of the neural networks for forecasting electron concentration profile 3 h in advance based on given local time and the Kp geomagnetic index. There have been a number of studies utilizing neural networks in a similar manner for upper atmosphere parameter forecasting. Much of the research has been concentrated on the forecasting of the critical frequency of the F2 layer (f_oF2). Several authors have presented studies of 1-h-ahead predictions of the f_oF2 (Altinay et al., 1997; Cander et al., 1998; Wintoft and Cander, 1999; Kumluca et al., 2000). Longer term (up to 24 h) predictions have also been carried out recently. Wintoft and Cander (2000) used a time-delay feedforward neural network with a back propagation learning algorithm to forecast the f_oF2 values 1–24 h in advance. Tulunay et al. (2000) used a model which involves spatial variations as well as temporal variations for the forecasting of f_oF2 up to 24 h in advance. The Tulunay and Ozkaptan model used a traditional feedforward network trained with the back-propagation algorithm. McKinnell and Poole (2001) used neural networks to produce a bottomside electron density profile. They also showed that neural networks are ideally suited tools to be used in quantifying the variability of ionosphere parameters under multivariate conditions. They demonstrated that for a given input specification, neural networks can be used to determine the range of expected parameter variations and to provide estimation for the unpredictable portion of parameter variability.

These studies demonstrated that neural networks provide a viable tool to study and model complex nonlinear phenomena characteristic of the ionosphere. The existence of nearly two solar cycles of ionosphere electron concentration profiles collected by the Arecibo Incoherent Scatter Radar (ISR) provided a rich set of

* Corresponding author. Tel.: +1 513 523 6070; fax: +1 513 529 1454.
E-mail address: Mortonyt@muohio.edu (J.D. Martin).

0273-1177/$30 © 2005 COSPAR. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.asr.2003.12.015

data that can be used to train a network model. Electron concentration profiles are functions of local time, seasonal changes, and solar activity. The geomagnetic index Kp is selected as the input parameter that reflects the seasonable and solar activity impact on the electron concentration distributions.

Two different neural network models were created in this project. One is the classical feedforward model. In this model, the 24-h local time period is divided into eight sections. Each section covers 3 h and has a corresponding Kp index associated with it. These resulting Kp indices serves as the eight inputs to the network.

Since each Kp index has a local time associated with it, local time became an embedded input parameter. The model output is a 15-element array that stores the electron concentration profile in the 150–600 km altitude range with 30 m altitude resolution. Experimental electron concentration data collected by the Arecibo ISR are used as training data for the network. The resulting network model is then tested and validated using additional experimental data. We averaged our experimental data profiles over a 3-h period to obtain a single profile. Our initial effort had a target goal of accurate prediction one step into the future, implying a prediction time of 3-h in advance.

An alternative model, the Elman recurrent model, was also developed as a comparative measure of the feedforward model. The Elman recurrent model is especially suitable for long-term forecasts that predict multiple steps into the future based on current information. Therefore, the Elman recurrent model will be the focus of our subsequent studies. We did not anticipate the Elman recurrent model would outperform the feedforward model for our single step prediction goal. Our testing and validation results based on the two models as shown in the following sections verify this.

MatLab with the Neural Network Toolbox (NNToolbox) add-on was utilized for the creation of the neural networks in this project.

2. Feedforward and Elman recurrent neural network model

Neural network architectures have been discussed in detail in the literature (Hagan et al., 1996; Haykin, 1999). A brief overview of the feedforward and Elman recurrent neural network models are provided here.

The feedforward neural network consists of an input layer, a hidden layer, and an output layer. All input data to the network are propagated through the layers from the input, to the hidden, and finally to the output layer without any feedback. Fig. 1 is a schematic of the feedforward model. $X_{\mathrm{KP}}(t), X_{\mathrm{KP}}(t-1), \ldots, X_{\mathrm{KP}}(t-7)$ are the Kp indices for the current time section and previous seven time sections. n is the number of neurons.

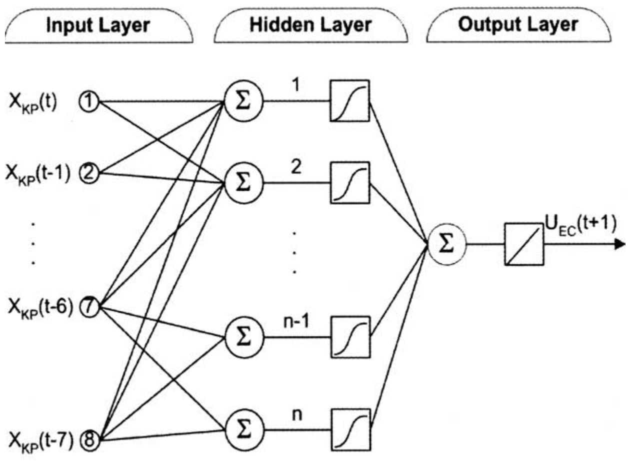

Fig. 1. Feedforward neural network architecture.

$U_{\mathrm{EC}}(t+1)$ is a 15-element array that contains the electron concentration profile at the next time section (i.e., 3 h later). A major advantage of the feedforward architecture is that it requires relatively low amounts of computing processing time during training. This is valuable when numerous simulations are necessary to evaluate a neural network performance.

Fig. 2 shows the schematics of the Elman recurrent model. The Elman recurrent model has an additional feedback loop. In our study, the 15 output values for a single electron concentration profile were used as the feedback to serve as the additional inputs for forecasting additional time steps into the future. $U_{\mathrm{EC}}(t+2)$, $U_{\mathrm{EC}}(t+3), \ldots, U_{\mathrm{EC}}(9+7)$ are the outputs representing the electron concentration profiles at $6, 9, \ldots, 21$ h in the future. This feedback connection allows the network to possess short term memory. The short term memory

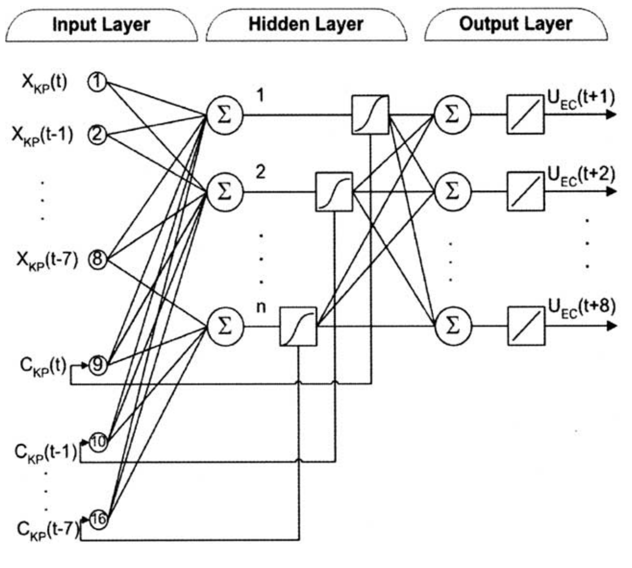

Fig. 2. Elman recurrent neural network architecture.

allows for temporal pattern recognition over a time-series data. The recurrent type of network is ideal for temporally dependent data and for multi-step predictions because of the recurrent connections that serve as additional inputs to predict the subsequent time steps in the future. Because of the more complex architecture of the recurrent model, however, there is a significant increase in training time compared with the feedforward model.

A logistic activation function Ψ, represented by Eq. (1) below, was used in both the feedforward and Elman recurrent models to apply a nonlinear connection between the input and output layers. This particular logistic function was chosen because it is the one most commonly used in a multilayered network. The variable b represents the slope parameter of the activation function which determines the shape of the curve

$$\Psi_j(n) = \frac{1}{1+e^{-bn}}. \tag{1}$$

3. Neural network training

Neural networks must be trained to map certain input patterns to output profiles. A neural network becomes trained by adjusting the weights (also referred to as free parameters) of the connections once these patterns are established through error interpretation. There are a variety of training algorithms that can be used to adjust the free parameters. The back-propagation algorithm is used in this project.

The back-propagation algorithm is an iterative learning rule which updates layer-to-layer weights based on the actual neural network outputs' deviation from a target. In each training cycle, or epoch, input data are fed through the system while free parameters are held constant. The outputs generated by the system are compared with a set of target output data. Through each epoch n of the training algorithm, the energy associated with the output error signal for each neuron j is calculated by the following equation:

$$E(n) = \frac{1}{2}\sum_{j=0}^{\text{Output Neurons}} (t_j(n) - y_j(n))^2, \tag{2}$$

where t is the target signal and y is the actual system output.

The induced local field v during training epoch n is defined as:

$$v_j(n) = \sum_{j=0}^{\text{Output Neurons}} \omega_{ij}(n) y_i(n). \tag{3}$$

E and v are used to calculate the local gradient δ, which determines the required direction of the changes in synaptic weights (ω)

$$\delta_j = -\frac{\partial E(n)}{\partial v_j(n)}. \tag{4}$$

Two parameters are critical for network performance, the learning rate η and the momentum rate μ. Both parameters may vary between 0 and 1. The learning rate is used to adjust the slope of the weight correction and it determines how long it will take the algorithm to converge to a minimum threshold root mean square (RMS) error. If the learning rate is too small, it will take the algorithm a relatively long time to converge. If the learning rate is too large, it is possible that the algorithm will diverge and never reach the threshold value. The learning rate is determined through trial and error and should be a reasonable combination of convergence and relatively low processing time. The weight adjustment calculations shown in Eq. (5) are used to update the free parameters of the system during each epoch

$$\Delta\omega_{ij}(n) = -\eta\frac{\partial E(n)}{\partial \omega_{ij}(n)}. \tag{5}$$

The momentum rate μ takes into account the weight-corrected magnitude and direction from the previous and current epoch. If the error gradient continuously moves toward the local minimum (i.e., the threshold value), the combination of the momentum term and the learning rate will increase the slope of the error surface, thereby accelerating its movement. If the error gradient fluctuates and repeatedly changes direction, the momentum term will help to smooth out these variations. With the introduction of the momentum term, the weight correction equation becomes that shown in Eq. (6). As a general rule of thump, a high momentum rate (>0.7) should be used in conjuction with a low learning rate (<0.3), and vice versa.

$$\Delta\omega_{ij}(n) = \eta\delta_i y_i + \mu\Delta\omega_{ij}(n-1). \tag{6}$$

Another important consideration in training is the selection of training data. A critical issue in data selection is the time coverage of the data set. It is essential to select data that spans several days or weeks that can result in a reasonably good generalization, but with no overfitting. The training data must also contain signals whose parameters are representative of a wide range of conditions. In our case, this means that the Kp index associated with the training data must have a range covering the normal geomagnetic condition, as well as disturbed cases.

4. Testing, validation, and results

The completion of the training algorithm optimized the synaptic weighted connections to establish reliable relationships between the inputs and outputs. During the testing and validation step, these free parameters are held constant while new inputs are fed to the network to produce a series of outputs. These outputs are then compared to a set of test data matching the actual produced outputs. If the actual output deviates from the target test set output above a threshold error value, the training parameters need to be adjusted and the neural network retrained.

In our testing, we varied learning rate and momentum rate from 0 to 1 with different training data time coverage (i.e., 3, 5, 10, 20, 50 days). Also, we varied the architecture by analyzing how the number of hidden neurons affected the accuracy of the networks.

The feedforward neural network's performance was evaluated based on over 10,000 automated simulations with varied architecture, training parameters, and training data time coverage. Fig. 3 is an example error analysis of a feedforward network with a learning rate of 0.05, momentum rate of 0.1, and minimum, average, and high error values over a set of varied hidden neurons of 1–100.

The error values for a simulation run were calculated by finding the average of the deviation of the target electron concentration value for a specific altitude (or output neuron) j against the actual output over the complete output electron concentration profile N. Our analysis shown that the number of hidden neurons had the largest effect on the error. With a fixed architecture, but varying training parameters and training data horizon, there was not a discernible pattern present to be able to determine if the training parameters had as large of an effect on the output error as did the number of hidden neurons.

Similar to that of the feedforward neural network, the Elman recurrent model performance was based on over 10,000 simulations with the same variation in architecture (number of hidden neurons) and training parameters. The error values were calculated in the same manner as for the feedforward model. Fig. 4 is an example result analysis of an Elman recurrent model with a learning rate of 0.05, momentum rate of 0.1, and minimum, average and high error values over a set of varied hidden neurons of 1–100. As with the feedforward model, we found that the number of hidden neurons had the largest effect on the system error. Direct comparison of Figs. 3 and 4 shows that the feedforward model has better performance, especially with an architecture having a small number of hidden neurons. Similar analysis was performed for additional training horizons of 6, 10, 15 and 20 days. We found that under every condition tested, the feedforward model is not only simpler and requires less computing power, it also provides more reliable results, if we are only interested in forecasting one step into the future. Considering that the inputs to the networks contains embedded time delay information, for one-step-into-the-future prediction, the history was already in the inputs. The additional memory associated with the Elman model is not needed.

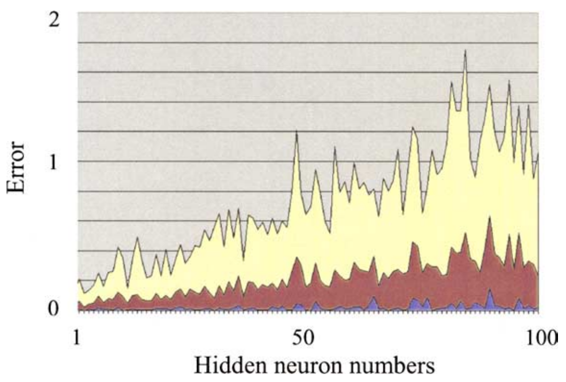

Fig. 3. Feedforward neural network min, avg, and max error vs. hidden neuron numbers.

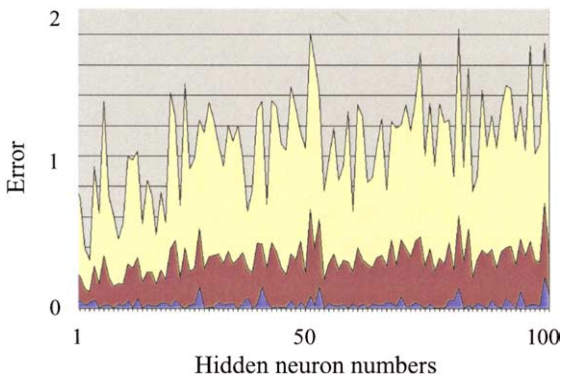

Fig. 4. Elman recurrent neural network min, avg, and max error vs. hidden neuron numbers.

A number of simulations were performed in order to determine the optimal training parameters, input data training horizon, and number of hidden neurons. The training parameters were held constant for 100 simulations, while the hidden neurons were repeatedly varied from 5 to 30. This was repeated through a number of different combinations of the learning and momentum rates. After over 5000 additional simulations, we found the following optimal architecture, training parameters, and training data time horizon:

We tested feedforward network with parameters in Table 1 over a set of 100 simulation using a set of validation data. Fig. 5 presents the training results of 16 such simulations. The results show that the trained

Table 1
Optimum training parameters

Learning rate (η)	Momentum rate (μ)	Training epochs	# of hidden neurons	Training data horizon
0.7	0.7	2000	16	3 days

Note that these optimum η and μ values are far from the ones used to generate Figs. 3 and 4.

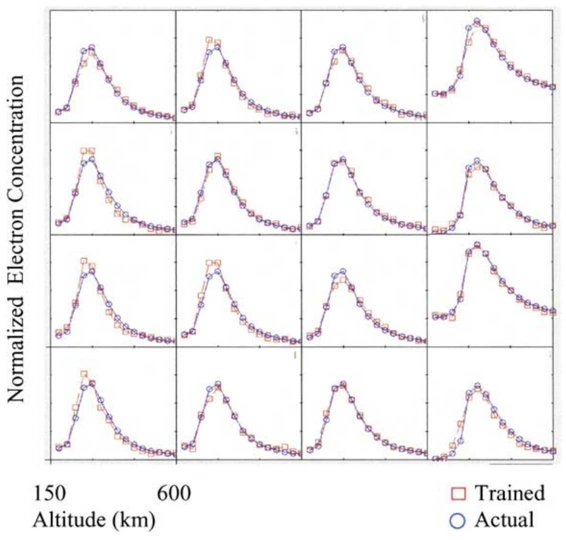

Fig. 5. Feedforward neural network training results with optimal setting.

neural network predicts electron concentration profiles very close to the actual ones. Our simulation shows that the optimal upper bound for the number of hidden neurons is 30.

After training each of these 100 neural networks, a simulation was performed in which the network was used to perform actual 3-h in advance predictions on real electron concentration profile measurements. For these simulations the neural network free parameters and architecture were fixed and provided with new input parameters to test the performance. Fig. 6 shows the actual performance. As can be seen from the figure, the feedforward neural network with optimal parameters predicts electron concentration profiles very close to the actual ones.

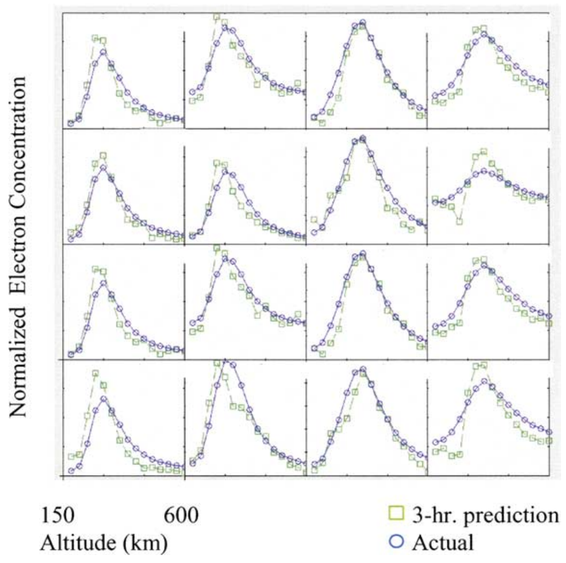

Fig. 6. Feedforward neural network prediction vs. actual results.

5. Conclusions

Our study found that the simple feedforward neural network was capable of providing reasonable forecasting of the electron concentration profiles in the upper atmosphere one step (3 h) into the future. The mean square error of over 100 simulations with a fixed set of training parameters, training data time horizon, and number of hidden nodes was found to be $11.42*10^{-4}$. Our network model was unsuccessful in forecasting electron concentration profiles during unique geomagnetic disturbances (i.e., solar flares, CME's, etc.). We believe this is due to the training method – the back-propagation training algorithm used to adjust the weighted average of the inputs which smoothes the data over a time horizon. This means, essentially, that any short-term/large-magnitude disturbances a few standard deviations from the mean would not affect the parameters.

We also found that the number of hidden neurons has the greatest effect on the neural network's performance. Zhang et al. (2001) found, however, that in addition to the number of hidden nodes, the number of input nodes had a significant effect on the neural network's predictive ability. They found that the number of input nodes has a much stronger effect on the performance than the number of hidden nodes in both in-sample fit and out-of-sample forecasting. Therefore, recommendations for future study would be to include variable input nodes during the performance evaluation simulations.

Additional future works include pre-process the ISR measurements in finer time resolution to reveal details of geomagnetic disturbances, to create neural network models that can forecast disturbance behavior, and to forecast multiple steps into the future. Recurrent type of network model will be reinvestigated for forecasting multi-steps into future. We would also like to continue to analyze different neural network architectures, different ranges of input and training data, along with analyzing the performance of the network under various conditions.

References

Altinay, O., Tulunay, E., Tulunay, Y. Forecasting of ionospheric critical frequency using neural network. Geophys. Res. Lett. 24, 1467–1470, 1997.

Cander, L.T., Stankovic, S., Milosvaljevic, M. Dynamic ionospheric prediction by neural networks. in: Scandahl, I., Jonsson, E. (Eds.), Proceedings of the Second International Workshop on Artificial Intelligence Applications in Solar-Terrestrial Physics, ESA WPP-148 Proceedings. European Space Agency, Paris, pp. 225–228, 1998.

Hagan, M.T., Demuth, H.B., Beale, M. Neural Network Desing. PWS Pub. Co., Boston, 1996.

Haykin, S. Neural Networks, A Comprehensive Foundation. Prentice Hall Inc., Upper Saddle River, NJ, 1999.

Kumluca, A., Tulunay, E., Topalli, I., Tulunay, Y. Temporal and spatial forecasting of ionospheric critical frequency using neural networks. Radio Sci. 34, 1497–1506, 2000.

McKinnell, L.A., Poole, A.W.V. Ionospheric variability and electron density profile studies with neural networks. Adv. Space Res. 27, 83–90, 2001.

Tulunay, E., Ozkaptan, C., Tulunay, Y. Temporal and spatial forecasting of the f_oF2 values up to twenty four hours in advance. Phys. Chem. Earth 25, 281–285, 2000.

Wintoft, P., Cander, Lj.R. Short-term prediction of f_oF2 using time-delay neural network. Phys. Chem. Earth 24, 343–347, 1999.

Wintoft, P., Cander, Lj.R. Ionospheric f_oF2 storm forecasting using neural networks. Phys. Chem. Earth 25, 267–273, 2000.

Zhang, G., Patuwo, B.E., Hu, M.Y. A simulation study of artificial neural networks for nonlinear time-series forecasting. Comput. Oper. Res. 28, 381–396, 2001.