November 2006 Volume 1 Issue 4-6

Trade Science Inc.

Environmental Science

An Indian Journal

Current Research Paper

ESAIJ, 1(4-6), 2006 [152-158]

Developing Methods To Train Neural Networks For Time-Series Prediction In Environmental Systems

Corresponding Author

Yongliang Shi
Research Center for Eco-Environment Sciences, Chinese Academy of Science
18 Shuangqing Road, Beijing, (CHINA)
E-Mail: shiyongliang@126.com

Received: 7th September, 2006
Accepted: 1st November, 2006

Web Publication Date : 14th November, 2006

Co-Authors

Liang Chen[1], Jin Liu[2], Yidong Lei[3]
[1]Research Center for Eco-Environment Sciences, Chinese Academy of Science 18 Shuangqing Road, Beijing, (CHINA)
[2] State Key Laboratory of Software Engineering, Wuhan University Luojiashan, Wuhan, 430072, (CHINA), typhoon2008@hotmail.com
[3] Department of Environment Sciences and Engineering, Fudan University 220 Handan Road, Shanghai, (CHINA), Restone@126.com

ABSTRACT

We present a novel method to training neural networks for predicting future variable values of environmental system. Time-series data including soil, streamwater and climatic variables were measured hourly over several month periods in two situations in Qingpu district, 45 kilometers west Shanghai city, using data loggers and other measuring instruments. The data sets were used to train neural networks using three different methods, including a novel, biologically plausible system. Temporal pattern recognition capabilities using each method were investigated. The novel method proved equally capable in predicting future variable values using large data sets as the other two methods. An argument is made for this method, named the 'Local Interaction' method, providing valid competition to other neural network and statistical methods in the detection of patterns and prediction of events in complex environmental systems.

KEYWORDS

Time-series prediction;
Neural networks;
Simulated annealing;
Backpropagation;
Ecological prediction.

INTRODUCTION

In many cases, environmental system situations are so complex and non-linear that useful models of their dynamics can only be obtained through empirical data, rather than through analytical equations. Statistical analysis of ecological system data sets can be difficult when confronted with systems in which massive amounts of data are generated, for example, through data loggers.

Over the past decade, increased computer power has made the implementation of various artificial intelligence (AI) theories practicable[1]. show that neural networks are appropriate in situations where

the underlying relationships are poorly known[2]. Investigated optimal structure for neural networks used to forecast chaotic time series, of which environmental data are a good example.

A common problem encountered when applying neural networks to the modeling of environmental systems is the architecture of the neural network. Even amongst researchers familiar with neural network use, the implementation of a known neural network to deal with a specific situation is often accompanied with a great deal of trial and error, usually in finding the correct number of nodes, the system of node connection or even the training algorithm to use[3,4]. discussed this problem and a possible system of solving it using a Bayesian model selection procedure. However, their method was only applicable to single-output neural networks and so cannot be applied to the multiple-input, multiple-output models required here. The novel method presented here, named the "Local Interaction" method, is intended to eliminate many of the problems inherent in neural network design.

A popular neural network algorithm is that of back-propagation, the principles of which are discussed in following sections. Here we apply a novel NEURAL NETWORK training method that is more biologically realistic, in that connection adjustments are based purely on the activations of sending and receiving nodes. This method is compared to the backpropagation method by applying both approaches to the analysis of time-variable environmental data, in an effort to predict both variable values and specific "events"[5]. The time-variable data used were obtained from data loggers attached to environmental sensors.

Data resource

A variety of methods were used to obtain environmental variable measures. In the field, standard data loggers and sensing equipment were used[6]. In each of the two situations, measurements were made of variables that were of specific interest to researchers and that were believed to be related to each other in some manner.

Dianpu river is a small stream that forms part of the huangpu river in west Shanghai and its backswamp is typical of many silt areas in the east China. The Dianpu river watershed has an altitude range of 3-12m and consists largely of reed swampland.

Waterstream was continually monitored at a fibreglass flume installed close to the outflow of Dianpu river. Discharge was recorded in every 5 min and data were downloaded from loggers at both flumes every 2 weeks. Stream temperature was also logged simultaneously with a temperature probe inserted below the stream surface close to the flume[7]. Precipitation was measured by tipping-bucket raingauges connected to loggers, located adjacent to the stream at altitudes of 3 m and 10 m. Ambient air and soil temperatures were also monitored continually using thermographs; soil thermographs were installed at the depths of 0.15-0.20m(shallow) and 0.45-0.50m (deep).

The Dianpu river data variables used were stream flow rate, stream temperature, precipitation rate, air temperature and soil temperature at a specific depth (50 cm). Measurements were made on an hourly basis during the months January-June 2004.

The weather mast was sited near Xiayang town, owned by Shanghai Meteorological Observatory (SMO). Parameters recorded were temperature (°C), relative humidity (%), solar radiation (W m^{-2}), wind speed (m s^{-1}) and wind direction (°). Data were recorded hourly, from the 1th of June, 2005 until the 15th of October, 2005.In both data sets, the information is organized so that all variables measured at a particular time are given as neural network inputs[6], followed by the next time step information and so on. The values of each variable are standardized to fit on a scale of 0 to 1 in order that neural networks would be able to make practical use of the information.

Neural network design and operation

For all neural networks used throughout this work, the weights of synaptic connection are in the range [-1, 1], and node activations are in the range [0, 1], with activation y given as the following function of input x:

$$y = \frac{1}{1+\exp(-cx)} \quad (3)$$

This gives the activation curve given in figure 1.

1.Backpropagation

The backpropagation training algorithm is an adaptation of another training algorithm, namely the delta rule[8,9]. If we suppose that the error between actual and target pattern for a particular output node is given as some function E =E(w1,w2,w3,...) of the synapse weights connected to that node, as shown in figure 1, then in order to minimize the error we have to find the lowest point on the curve. For any function y=y(x) we have

$$dy = dx(\frac{dy}{dx}) \quad (4)$$

If we put

$$dx = -\alpha(\frac{dy}{dx}) \quad (5)$$

with $\alpha>0$, we then get

$$dy = -\alpha(\frac{dy}{dx})^2 \quad (6)$$

which implies that we have reduced the function by stepping down the slope. This process can be repeated until the local minimum is reached.

In standard practice, the error on node j is defined as the squared difference between the actual and the target values:

$$E_j = (t_j - a_j)^2 \quad (7)$$

which gives us

$$\frac{dE_j}{da_j} = -2(t_j - a_j) \quad (8)$$

We know that

$$a_j = f(\sum_i w_{ij} a_i) \quad (9)$$

If we ignore the general case for the moment and assume simply that

$$a_j = \sum_i w_{ij} a_i \quad (10)$$

$$\frac{dE_j}{dw_{ij}} = -2(t_j - a_j)a_i \quad (11)$$

with limits at (-1) and (1), then we obtain
If we substitute dw_{ij} for dx and dE for dy in Eq. (5) we get

$$dw_{ij} = \alpha(t_j - a_j)a_i \quad (12)$$

If we keep the *f* (given in Eq. (3)) in Eq.(9) and work out Eq.(12) again, we obtain the completed delta rule:

$$\Delta w_{ij} = \alpha c a_j(1 - a_j)(t_j - a_j)a_i \quad (13)$$

The extra terms here, $c\alpha$, (1 - α) are collectively given the symbol σ_j.

The mapping of an input layer onto an output layer is often sufficient for the system being modeled. However, some systems are of sufficient complexity that additional layers are required. The delta rule works for two-layer neural networks, but cannot be applied directly to neural networks with hidden layers because it is necessary to know the target value for each node. For hidden layer nodes, the target value cannot be predicted.

This problem is resolved by first calculating the error on the output layer nodes and then using these values to determine the error (t_j - α_j) for hidden layer nodes. This is where the term backpropagation comes from. Each of the hidden nodes i is responsible for activating a node j in the next layer, for which the error has already been calculated. The total error for i equals the sum of $w_{ij}\delta_j$, where j includes all of the nodes that i is responsible for activating and d is the error of node j. Node i is therefore responsible for a certain amount of error in later nodes, an amount that is used as its own error definition. So, the alteration to synapse w_{ij} is given by

$$\Delta w_{ij} = \alpha\sigma_i(\sum_k \delta_k w_{jk})a_i \quad (14)$$

and the summation $\sum_k \delta_k w_{jk}$ equals δ_j for calculation of the next layer of alterations. For each of the three data sets used, the backpropagation network was composed of the input layer, two hidden layers of 50 nodes each (fully connected) and the output layer. The value of α was set at 0.01, c was set at 5 through a trial-and-error process, and the network was trained until the average error over the last 1000 steps no longer decreased.

2.Local interaction method

The backpropagation method is the most popularly used neural network training method, particularly in situations where a transformation from input to output sets is required, for example, value classi-

fication. However, it relies heavily on the biologically implausible assumption that individual neurons in hidden layers of a network can have knowledge of the error associated with their activation. Biologically plausible models of the nervous system state that neurons have no individual memory or awareness of their actions, and that their behavior is dependent only upon their own activation history and the activation of nodes to which they have synaptic connections.

For the backpropagation method, the value of c for each node is fixed at some predetermined constant. Higher (lower) values of c make the activation curve flatter (steeper) around the zero point, as can be interpreted from figure 1. The activation curve has a higher gradient nearer to the zero point, and so in the novel method the value of c is not fixed, but can be adjusted. Using the same activation curve as in the backpropagation method (Eq.(1)), this adjustment takes the form

$$\Delta c_i = \beta(0.25 - |0.5 - a_i|) \quad (15)$$

and is shown in figure 2. The value of β is set at 0.01. The synapse adjustment algorithm is designed to alter synapses in such a way that their weighting reflects the importance of their connection. For two nodes that are strongly coupled the synapse will be stronger, while for two nodes whose activations have little correlation the synapse will be weak. Because of this, synaptic reinforcements depend upon the difference between a node activation and its expected activation due to the last several steps.

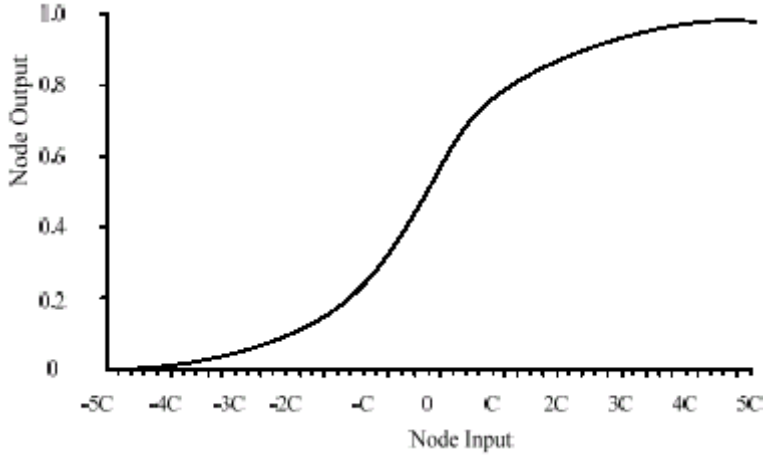

Figure 1: Activation curve for neural network nodes

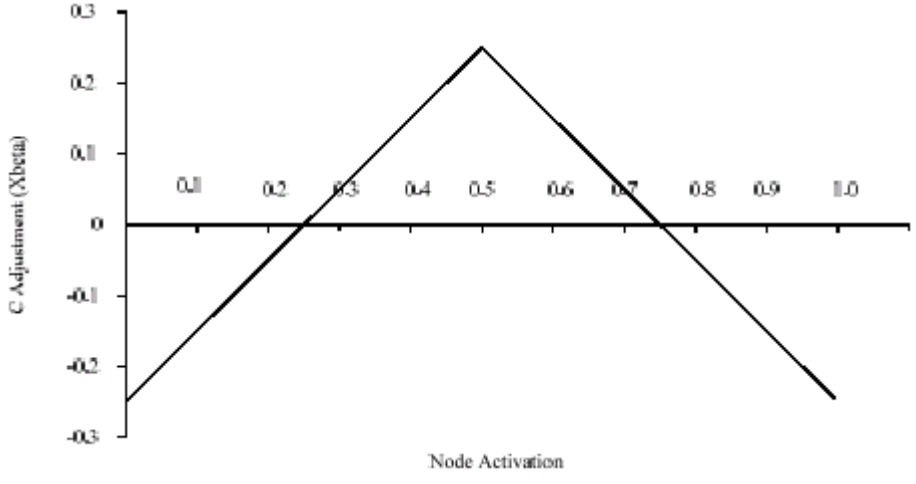

Figure 2: Adjustment curve for c activation constant

The expected activation $\tilde{a}$ for a particular node at time step t is calculated using Eq.(16):

$$\tilde{a}_t = \tilde{a}_t - 1 + 0.1(a_{t-1} - \tilde{a}_{t-1}) \quad (16)$$

The difference between actual and expected node activation $(a - \tilde{a})$ is calculated for both the sending i and receiving j node at the end of each synapse, and the synaptic adjustment Δw_{ij} is given by the following equations:

$$\Delta w_{ij} = \alpha[(a_i - \tilde{a}_i) + (a_j - \tilde{a}_j)] \quad (17a)$$

$$(\alpha_i - \bar{\alpha}_i > 0, \alpha_j - \bar{\alpha}_j > 0)$$

$$\Delta w_{ij} = \alpha[-(a_i - \tilde{a}_i) + (a_j - \tilde{a}_j) \quad (17b)$$

$$(\alpha_i - \bar{\alpha}_i > 0, \alpha_j - \bar{\alpha}_j < 0)$$

$$\Delta w_{ij} = \alpha[(a_i - \tilde{a}_i) - (a_j - \tilde{a}_j) \quad (17c)$$

$$(\alpha_i - \bar{\alpha}_i < 0, \alpha_j - \bar{\alpha}_j > 0)$$

$$\Delta w_{ij} = \alpha[-(a_i - \tilde{a}_i) - (a_j - \tilde{a}_j) \quad (17d)$$

$$(\alpha_i - \bar{\alpha}_i < 0, \alpha_j - \bar{\alpha}_j < 0)$$

where a is set to 0.01.

3. Simulated annealing

In a manner similar to that of backpropagation, a method of reducing the error between actual and predicted outputs is sought. However, this method does not rely on adjusting the synaptic weights according to the measured error, but rather uses a stochastic method to gradually improve the performance of the system towards some optimal state. The simulated annealing method as applied here is best described using the following algorithm:

1. Measure the fitness of the system.
2. Adjust some parameter of the system by a relatively small amount.
3. Again measure the fitness of the system (error between actual and predicted values).
4. If the fitness has improved then retain the change. If not, discard it.
5. Return to 1.

In the case used here, the system will be a neural network with the same topology and behaviour as that used in the backpropagation section, but with a different method of synaptic adjustment. Synaptic weights will be initially randomized, and will be the 'parameter' of the system that is adjusted by a small level. In each loop of the algorithm above, one synapse will be selected at random and adjusted by ±0.01, with the sign also selected at random. In order to avoid the system settling into some local minimum, the adjustment will be retained 1% of the time even if the fitness of the system (measured as the mean prediction error) does not improve.

There are parallels here with genetic algorithms, in the use of more than one input parameter and the definition of system 'fitness'. Indeed it is felt here that the method as it has been applied is a combination of simulated annealing and GAs, as the connections are being treated as components of the system that are subject to evolutionary pressures while the overall system is being treated as a case problem in minimizing error or 'energy' levels. However, as GA methods rely on the use of selection from a pool of candidates, it is felt that attributing the implementation to simulated annealing is more accurate.

RESULTS

For each of the three NN methods, five repetitions were made for each of the two data sets. In each case, the network was adjusted until system performance (averaged over the last 1000 training steps) stopped improving. In the case of the backpropagation system this took approximately 15000 steps, while for the local interaction and simulated annealing the number of steps taken were 5500 and 420000 steps, respectively.

Dianpu river

Figure 3 gives the histogram of prediction errors for the Dianpu River data, allowing comparison of the three methods. For the simulated annealing and Local Interaction methods, over 50% of the predictions were within 10% of the correct values, and over 90% were within 30%. For backpropagation, the results were less accurate (28% within 10%, 51%

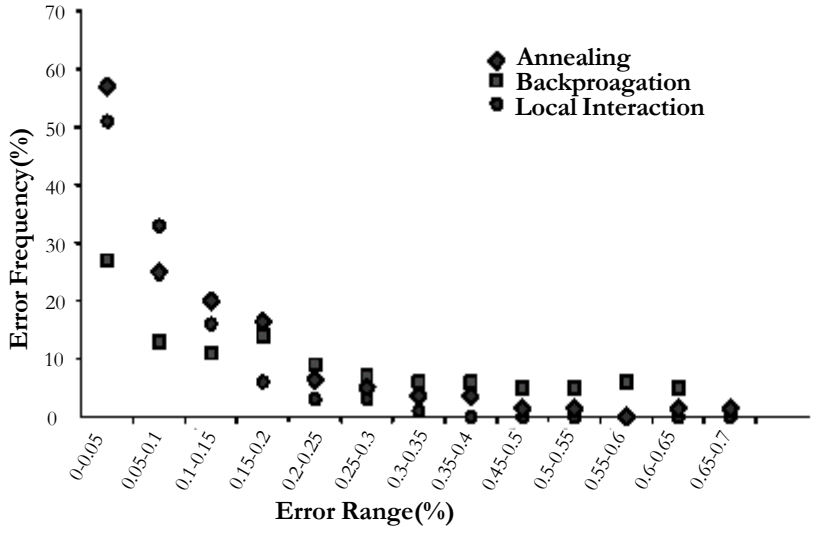

Figure 3: Error histogram for Dianpu River data set

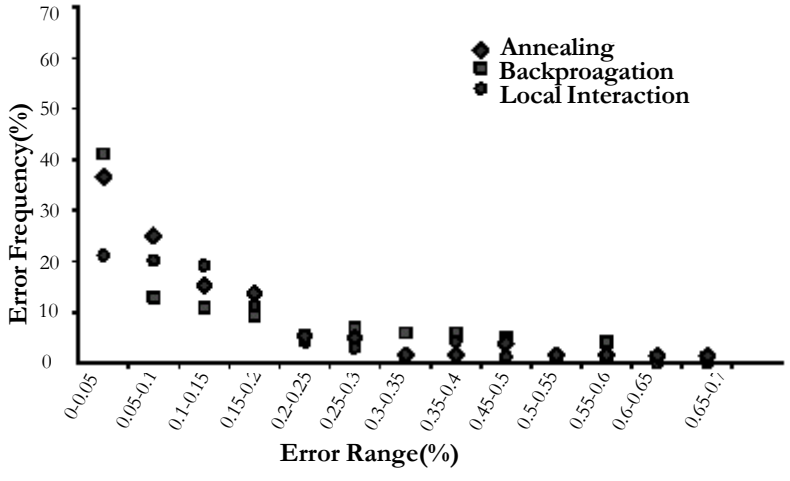

Figure 4: Error histogram for Xiayang town data set

within 30%).

Xiayang town

Figure 4 gives a comparable histogram of prediction errors for the Xiayang town data set, showing similarities and differences with the Dianpu River data set. Here, the annealing and backpropagation methods prove more accurate than the Local Interaction method, although for the novel method 50% of the predicted values are within 20% accuracy.

DISCUSSION

A novel neural network training method has been developed that is more biologically plausible than the traditionally used methods of backpropagation or simulated annealing. This method, when applied to complex system time series data sets, gives comparable results to the two more traditional methods used. Furthermore, the Local Interaction method is a strong contender for automatic temporal attractor development and retrieval for two reasons.

The final product of the evolution and training methods is a neural network-based model specific to each investigated situation, in which the user's given inputs result in predicted values for the variables 1 h ahead. These outputs can then be used as inputs for another prediction cycle, allowing the user to obtain predictions for further forward in time.

Given that the predictions for a single hour ahead are not and never will be 100% accurate, the reliability of the system will be lower for predictions in the long rather than the short term. It is interesting to note that if the system's long-term predictions for many different sets of starting values are compared, there is no tendency towards a stable "attractor". Rather, the system behaves chaotically, with variable predictions covering the entire range of possibilities.

While it is fair to say that research using Neural networks is already far removed from biological plausibility due to oversimplification or the use of contrived mathematical methods, it is also fair to say that attempting to maintain biological plausibility may allow researchers to compare the abilities of their systems against the tried and tested reliability of organic learning systems. It is capable of predicting the behavior of a system over several time steps, given only starting values. Non-feedback methods such as backpropagation are not capable of this, and whereas a simulated annealing method may allow a temporal sequence recognition system to work, it will lack the flexibility of the Local Interaction method. The Local Interaction method is a more efficient training technique in that (a) it is more rapidly trained than the other two methods, and (b) it avoids the "local minima" problem that the others have, by not attempting to minimize errors in output. This new method shows good potential, but needs to be evaluated further. Future work includes carrying out the training procedures used here on other data sets to further compare their abilities. Also, data sets containing non-temporal data will be used to compare the various methods, and to investigate the further applicability of each.

One important consideration is that of solving the 'black-box' nature of neural networks, as described. The methods they describe for analyzing trained neural networks would give insights into which factors are most important, and how the factors interact. Neural networks can therefore provide a method of modeling environmental systems, either as a 'black-box' system or through subsequent analysis of trained networks. Spatial as well as temporal analysis can be performed by such trained models, and may provide insights which classical statistical analysis methods cannot. Further work is being carried out on prediction of plant array structural features using a variety of NEURAL NETWORK methods.

ACKNOWLEDGEMENTS

Thanks to the Shanghai Meteorological Observatory (SMO) providing precipitation data. Additional thanks go to Dr. Chenmin (Wuhan University) for comments and suggestions.

REFERENCES

[1] K.A.De Oliveira, A.Vannucci, E.C.da Silva; Physica A, **284**, 393-404 **(2000)**.
[2] C.M.Bishop; 'Neural Networks for Pattern Recognition', Clarendon Press, Oxford, 112-126 **(1995)**.
[3] G.Boudjema, N.P.Chau; Ecol.Model, **91**, 15-23 **(1996)**.
[4] R.Tsaih, Y.Hsu, C.C.Lai; Decision Support Syst., **23**, 161-174 **(1998)**.
[5] J.Zhang, A.J.Morris, E.B.Martin; Comput.Chem.Eng., **22**, 1051-1063 **(1998)**.
[6] C.M.Zealand, D.H.Burn, S.P.Simonovic; J.Hydrol., **214**, 32-48 **(1999)**.
[7] M.Zhang, J.Fulcher, R.A.Scofield; Neurocomputing, **16**, 97-115 **(1997)**.
[8] Shi Zhongzhi; 'Artificial Life' (in Chinese), Scientific Publishing House, Beijing, China, 102-118 **(2001)**.
[9] M.Batty; GeoComputation Using Cellular Automata. In S.Openshaw, and R.J.Abrahart (Eds.), 'Geo Computation', London, UK, Taylor & Francis, 95-126 **(2000)**.
[10] J.Zhang; Neurocomputing, **25**, 93-113 **(1999)**.
[11] J.Sjoberg, Q.Zhang, L.Ljung, A.Benveniste, B.Delyon, P.Y.Glorennec, H.Hjalmarsson, A.Juditsky; Automatica, **31**, 1691-1724 **(1995)**.