A novel method for training neural networks for time-series prediction in environmental systems

M.J. Aitkenhead a,*, A.J.S. McDonald b, J.J. Dawson b, G. Couper b, R.P. Smart b, M. Billett b, D. Hope b, S. Palmer b

a The Macaulay Land Use Research Institute, Craigiebuckler, Aberdeen AB15 8QH, Scotland, UK
b Department of Plant & Soil Science, University of Aberdeen, Aberdeen AB24 3JU, Scotland, UK

Received 6 March 2002; received in revised form 23 October 2002; accepted 20 November 2002

Abstract

Soil, streamwater and climatic variables were measured hourly over several month periods in two situations in North-East (NE) Scotland, using data loggers and other measuring instruments. One of the locations was on agricultural land near Inverness while the other was at an acidic peat moorland site in the River Dee catchment. 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, termed 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 biological systems.

© 2002 Elsevier Science B.V. All rights reserved.

Keywords: Stream dynamics; Time-series prediction; Neural networks; Simulated annealing; Backpropagation; Environmental prediction

1. Introduction

In many cases, environmental situations of interest are sufficiently 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 environmental 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. Lek and Guégan (1999) show that neural networks (NNs) are appropriate in situations where the underlying relationships are poorly known. De Oliveira et al. (2000) investigated the optimal structure for NNs used to forecast chaotic time series, of which environmental data are a good example. Zhang et al. (1997) in using NNs to provide rainfall estimation, showed that a generalisation of neural network theory allows any piecewise continuous function to be approximated. This implies that it is possible to use NNs to develop any transformation function. So, if there is a well-behaved relationship between a set of input values and a set of

* Corresponding author. Tel.: +44-1224-498200; fax: +44-1224-311556. E-mail address: m.j.aitkenhead@abdn.ac.uk (M.J. Aitkenhead).
output values, that relationship can be expressed using neural networks.

Barciela et al. (1999) used neural networks to successfully predict the short-term temporal behaviour of phytoplankton biomass in a coastal region, achieving particularly high correlations between predicted and actual values over daily time increments. Karul et al. (2000) give examples of the use of predictive models for complex systems, including modelling of the eutrophication process of three large water bodies in Turkey.

A common problem encountered when applying neural networks to the modelling of biological systems is the architecture of the NN. Even amongst researchers familiar with NN use, the implementation of a known NN 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. Vila et al. (1999) 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 NN design.

A popular neural network algorithm is that of backpropagation, the principles of which are discussed in following sections. Zealand et al. (1999) applied backpropagation to forecasting stream flow. Tsaih et al. (1998) used the same principle as part of a system for predicting stock market values. Although highly successful in many aspects of system prediction and control, this algorithm is not biologically plausible, relying as it does on determining the ‘target’ levels of activity for nodes in hidden layers between the input and output layers. Here we apply a novel NN 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’. The time-variable data used were obtained from dataloggers attached to environmental sensors.

2. Environmental data capture

A variety of methods were used to obtain environmental variable measures. In the field, standard data loggers and sensing equipment were used. 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.

Brocky Burn is a small headwater stream (2.7 km in length; 1.30 km² catchment area) that forms part of the larger River Dee catchment in NE Scotland and is typical of many acidic peatland areas in the UK. The Brocky Burn catchment has an altitude range of 250–549 m and consists largely of heather moorland, which is managed for grouse by periodic burning. The stream itself is flashy and rocky with many riffles, pools and small waterfalls.

Discharge was continually monitored at a fibreglass flume (1.75 m × 1.44 m × 0.60 m, SEPA) installed close to the outflow of Brocky Burn (NO615832). The height of water in the flume was continuously measured using a pressure transducer connected to a datalogger (Campbell Scientific). The pressure transducer was calibrated with an upstream gaugeboard, positioned so that the zero mark on the gaugeboard was level with the base of the upstream end of the flume. The flume rating equations linking water height to discharge were as follows:

\[ Q = 1.10028 \times H^{1.12}, \quad \text{for } H < 0.255 \]  
\[ Q = 1.31732 \times H^{2.24}, \quad \text{for } H > 0.254 \text{ and } < 0.610 \]

where \( Q \) is the discharge (m³ s⁻¹) and \( H \) is the head of water (m).

Discharge was recorded in every 5 min and data were downloaded from loggers at both flumes every 2 weeks. The data were then processed to produce hourly discharge values. Stream temperature was also logged simultaneously with a temperature probe inserted below the stream surface close to the flume. Precipitation was measured by tipping-bucket rain gauges connected to loggers (Campbell Scientific), located adjacent to the stream at altitudes of 250 m (SEPA, unpublished data) and 405 m. Ambient air
and soil temperatures were also monitored continually using thermisters; soil thermisters were installed at the depths of 0.15-0.20 (shallow) and 0.45-0.50 m (deep).

The Brocky Burn 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 1997. The total data set size is 3530 measurements of the five variables, with neural network inputs being the measured variable values at time \( t \) and target outputs being the values measured at time \( t + 1 \).

The weather mast was sited in Lagoon field (OS NH3776-8506), owned by TIO Ltd., Culblair Farm, Dalcross, Inverness, Scotland. 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 8th of June, 2000 until the 24th of October, 2000, using a four channel Data Hog (Skye Instruments) with integral RH sensor and temperature sensor (SDL 5260) with the wind speed and direction sensors (WSD 1) and a pyranometer (SKS 1110/0) also connected. The instruments were secured to a portable 2 m mast with metal base-plate and guy ropes (ACC/11C) and powered using a solar panel power source (ACC/5). All monitoring equipment was purchased by Aberdeen University Centre for Organic Agriculture from Skye Instruments Ltd., Unit 32, Ddole Industrial Estate, Llandrindod Wells, Powys, Wales.

In both data sets, the information is organised so that all variables measured at a particular time are given as neural network inputs, 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.

3. 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 \( x \):
\[
y = \frac{1}{1 + \exp(-cx)}
\]
This gives the activation curve given in Fig. 1.

3.1. Backpropagation

The backpropagation training algorithm is an adaptation of another training algorithm, namely the delta rule. If we suppose that the error between actual and target pattern for a particular output node is given as some function \( E = E(w_1, w_2, w, \ldots) \) of the synaptic weights connected to that node, as shown in Fig. 1, then in order to minimise the error we have to find the lowest point on the curve. For any function \( y = y(x) \) we have
\[
dy = dx \left( \frac{dy}{dx} \right)
\]
If we put
\[
dx = -\alpha \left( \frac{dy}{dx} \right)
\]
with \( \alpha > 0 \), we then get
\[
dy = -\alpha \left( \frac{dy}{dx} \right)^2
\]
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:
\[
E_j = (t_j - a_j)^2
\]
which gives us
\[
\frac{dE_j}{da_j} = -2(t_j - a_j)
\]
We know that
\[
a_j = f \left( \sum \limits_i w_{ij} a_i \right)
\]
If we ignore the general case for the moment and assume simply that
\[
a_j = \sum \limits_i w_{ij} a_i
\]
with limits at \((-1)\) and \((1)\), then we obtain
\[
\frac{dE_j}{dw_{ij}} = -2(t_j - a_j)a_i
\] (11)

If we substitute \(dw_{ij}\) for \(dx\) and \(dE_j\) for \(dy\) in Eq. (5) we get
\[
dw_{ij} = a(t_j - a_j)a_i
\] (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:
\[
\Delta w_{ij} = ca_j(1 - a_j)(t_j - a_j)a_i
\] (13)
The extra terms here, \(ca_j(1 - a_j)\) are collectively given the symbol \(\sigma_j\).

The mapping of an input layer onto an output layer is often sufficient for the system being modelled. However, some systems are of sufficient complexity that additional layers are required. The delta rule works for two-layer NNs, 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 \((t_j - a_j)\) for hidden layer nodes. This is where the term \(\text{backpropagation}\) 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 \(\delta_j\) 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
\[
\Delta w_{ij} = a\sigma_i \left( \sum_k \delta_j w_{jk} \right) a_i
\] (14)
and the summation \(\sum_k \delta_j w_{jk}\) equals \(\delta_j\) 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 \(\alpha\) 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.

A known problem with the backpropagation training method is that it searches for a local error minimum.
rather than a global one, in an error space that may have many local minima (Boudjema and Chau, 1996). Introducing noise to the system has been suggested as a means of forcing the system to ‘jump’ out of the local minimum that it is in, but there are difficulties in optimising the noise level required. Noise is not introduced in this method.

3.2. 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 classification. 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 behaviour is dependent only upon their own activation history and the activation of nodes to which they have synaptic connections. The importance of biological plausibility is taken further in Section 5.

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 Fig. 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

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

and is shown in Fig. 2. The value of $\beta$ 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. The expected activation $\tilde{a}$ for a particular node at time step $t$ is calculated using Eq. (16):

$$\tilde{a}_i = \tilde{a}_{i-1} + 0.1(a_{i-1} - \tilde{a}_{i-1})$$

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 $\Delta w_{ij}$ is given by the following equations:

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

$$\begin{cases} a_i - \tilde{a}_i > 0, a_j - \tilde{a}_j > 0 \end{cases}$$

(17a)
\[ \Delta w_{ij} = \alpha \left[ (a_i - \tilde{a}_i) + (a_j - \tilde{a}_j) \right] \]  
\[ (a_i - \tilde{a}_i > 0, a_j - \tilde{a}_j < 0) \]  
\[ \Delta w_{ij} = \alpha \left[ (a_i - \tilde{a}_i) - (a_j - \tilde{a}_j) \right] \]  
\[ (a_i - \tilde{a}_i < 0, a_j - \tilde{a}_j > 0) \]  
\[ \Delta w_{ij} = \alpha \left[ -(a_i - \tilde{a}_i) - (a_j - \tilde{a}_j) \right] \]  
\[ (a_i - \tilde{a}_i < 0, a_j - \tilde{a}_j < 0) \]  

where \( \alpha \) is set to 0.01.

The first two aspects of the hybrid system were concerned with individual nodes and synapses. A third aspect is used to control the overall behaviour of the network. For a network in which excitation and inhibition are equal, the average level of activation for each node should be around 0.5. A bias towards excitatory or inhibitory synapses could lead to situations where the network is saturated with nodes of high or low activation, instead of having a wide range of activations. In order to counteract this, a measure of the average activation within the entire system, \( a_m \), is taken and synapses are adjusted to counteract low or high means. Eq. (18) gives the relationship concerned:

\[ \Delta w_{ij} = \beta \left( 0.5 - a_m \right) \]  

As in Eq. (15), the value of \( \beta \) is set at 0.01. Errors are calculated by determining the difference between the output values and the correct values from the data set, after which the output nodes are set to these correct values and the synaptic and nodal values are adjusted.

For this method, the network structure is that of three layers. The first layer is composed of the input set, the second is the processing layer (50 nodes) and the third is the output layer. The first layer feeds forward to the second layer, which feeds to both itself and the output layer in a fully connected manner.

During operation, the node values are not initialised to zero before each input as for the backpropagation method. This allows temporal sequence attractors to be developed.

### 3.3. 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 randomised, 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.

An investigation was made into the potential of Monte Carlo methods for providing an optimal system. This method relies on using a large number of systems in which the relevant values are selected randomly, thus providing a rapid way of exploring the phase space of the system. However, it was found that with the number of connections within the neural network, the phase space was so large that this method did not provide as rapid an approach to an optimal system as the simulated annealing system.

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 minimising error or ‘energy’ levels. However, as GA methods rely on the use of selection from a pool of candidates.
Fig. 3. Error histogram for Brocky Burn data set.

Fig. 4. Error histogram for Inverness data set.
it is felt that attributing the implementation to simulated annealing is more accurate.

4. 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 15,000 steps, while for the local interaction and simulated annealing the number of steps taken were 5500 and 420,000 steps, respectively. However, the time taken for each step varied according to the method, as each method involves different calculations. In terms of time taken on a 300 MHz PII PC, the time taken for completion of each method was as follows: backpropagation – 1100 s; Local Interaction method – 450 s; simulated annealing – 4200 s.

4.1. Brocky Burn

Fig. 3 gives the histogram of prediction errors for the Brocky Burn 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% within 30%).

4.2. Inverness

Fig. 4 gives a comparable histogram of prediction errors for the Inverness data set, showing similarities and differences with the Brocky Burn 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.

5. 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 NNs 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 behaviour 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 minimise 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 by Olden and Jackson (2002). The methods they describe for analysing 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 modelling 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 NN methods.

Acknowledgements

Thanks to the landowners (Fasque Estates) for access to the Brocky Burn study site and to Derek Fraser from SEPA for providing precipitation data. Additional thanks go to Dr. Pete Smith (University of Aberdeen) for comments and suggestions.

References


