A bounded exploration approach to constructive algorithms for recurrent neural networks

Romuald Boné, Michel Crucianu, Gilles Verley, Jean-Pierre Asselin de Beauville

A Bounded Exploration Approach to Constructive Algorithms for Recurrent Neural Networks

Romuald Boné, Michel Crucianu, Gilles Verley, Jean-Pierre Asselin de Beauville

Laboratoire d'Informatique
École d'Ingénieurs en Informatique pour l'Industrie
64 avenue Jean Portalis, 37200 Tours, France
{bone, crucianu, verley, asselin}@univ-tours.fr

Abstract: When long-term dependencies are present in a time series, the approximation capabilities of recurrent neural networks are difficult to exploit by gradient descent algorithms. It is easier for such algorithms to find good solutions if one includes connections with time delays in the recurrent networks. One can choose the locations and delays for these connections by the heuristic presented here. As shown on two benchmark problems, this heuristic produces very good results while keeping the total number of connections in the recurrent network to a minimum.

1. Introduction

Time series prediction has important applications in various domains: medicine, ecology, meteorology, industrial control, finance, etc. The most common approach to building a prediction model is to consider a fixed number of the past values of one or several time series (i.e. a time window of fixed size) and look for a function which provides the next value of the target series. In univariate regression, for instance, one is usually searching for the function f which gives the best estimate of the future value of the time series according to $\hat{x}_t = f(x_{t-1}, \ldots x_{t-p})$, p being the size of the time window. Such a model is called Nonlinear AutoRegressive (NAR).

Multilayer perceptrons (MLP, [1], [2]) can easily implement nonlinear functions f : one is simply using a network having an input layer of size p, enough neurons with sigmoidal activation functions in the hidden layer, and a single output neuron which is usually linear. Universal approximation results for feed-forward neural networks [3], [4] show that very general NAR can then be obtained. Also, it is easy to take into account past activation values of the hidden neurons by including new time windows for them.

Finite impulse response (FIR) connections put forward in [5] and [6] are an alternative method for defining time windows. A FIR connection is composed of a set of simple connections, encompassing a whole range of delays. Every component connection has its own weight. Replacing the simple connections of an MLP by FIR connections produces composite nonlinear autoregressive models which give better results on several reference benchmarks [7]. However, the resulting models may have too many degrees of freedom and pruning techniques are then employed for complexity control as in [8]. Also, variable selection methods can help in reducing the number of connections (delays) between the input and the hidden layer [9], [10]. We can also mention a new method, inspired by back-propagation, which was proposed in [11] for learning the number of delays associated to the existing connections of an MLP.

However, the use of MLPs for time series prediction has inherent limitations, since one cannot find an appropriate finite NAR model for every dynamical system. Even when such a model exists, it may have a very large number of parameters and behave poorly on new data. Recurrent neural networks (RNN) possess an implicit internal memory and do no longer need time windows in order to take into account the past values of the time series. RNNs prove to be significantly more powerful than feed-forward networks, as shown in [12], [13] or [14]. Unfortunately, the gradient descent algorithms which are commonly used for training RNNs [1], [15] have several weaknesses, the most important one being the difficulty of dealing with long-term dependencies in the time series [16].

Adding connections with time delays to the RNN often allows gradient descent algorithms to find better solutions in these cases [17], [18], [19]. Indeed, by acting as a shortcut between two distant moments, such a connection has a linear long-range contribution, with beneficial effects on the expression of the gradient of the error. But in the absence of prior knowledge concerning the problem to solve, how can one choose the locations and the delays associated to these new connections? By systematically replacing simple connections by FIR connections one obtains again oversized networks which are slow to train and have poor generalization abilities. Various

regularization techniques are then employed in order to improve generalization and this further increases the computational cost.

2. Bounded exploration for the addition of time-delayed connections

Constructive approaches for adapting the architecture of a neural network are usually more economical. An algorithm for the addition of time-delayed connections to recurrent networks should start with a simple, ordinary RNN and progressively add new connections according to some heuristic. Various factors can guide the choice of a heuristic, such as the prior knowledge related to the problem, the features of the learning algorithm, or the computational overhead. The time series we are interested in are characterized by medium or long-term dependencies, we prefer heuristics which are relatively general with respect to the detailed computation performed by the learning algorithm, and we require a low computation cost.

The heuristic we retained is a breadth-first search which can be summarized as follows: we explore all the alternatives for the location and the delay associated to a new connection by adding that connection and performing a few iterations of the learning algorithm; we keep the connection which produces the largest increase in performance during these learning steps. If the RNN we start with does not account well for the medium or long-term dependencies in the data, and these dependencies are not too complex, then by adding the appropriate connection we are likely to obtain right away a significantly lower error.

We can now give a more detailed outline of the constructive method we developed. In the experiments we performed we employed Back-Propagation Through Time (BPTT, [11]) as a learning algorithm. Learning by BPTT ends when error increases on a stop set, different from the learning set. At that moment we explore all the alternative RNN obtained from the previous one by adding a single connection: the weight of this connection is initialized to 0 and we leave the other weights in the RNN unchanged; we perform a limited number of learning steps using BPTT; we retain the connection which produces the largest decrease in error on the stop set during these learning steps. Note that error remains the same when a new connection with a weight of 0 is added. We then continue learning by BPTT on this new architecture; as soon as the error begins to increase on the stop set, we attempt to add a new connection – the whole process is reiterated. We call the resulting constructive algorithm Exploratory Back-Propagation Through Time (EBPTT). The algorithm eventually ends when the error on the stop set no longer decreases upon addition of a new connection, or a bound on the number of new connections is reached.

Several new parameters are required for this constructive algorithm: the maximal value for the delays of the new connections, the maximal number of new connections and the number of BPTT steps performed for each candidate connection during the exploratory stage. In choosing the value of the first parameter one should use prior knowledge related to the problem. If such information is not available we can rely on simple, linear measures such as self or cross-correlations to find a bound for the long-term dependencies. Note that this parameter is not the same as the maximal order of a FIR connection: indeed, when we add a connection of delay p, we do not simultaneously add $p-1$ connections with delays between 1 and $p-1$.

Computational cost governs the choice of the other two parameters. However, the experiments we present in the following show that the contribution of the new connections diminishes quickly as their number increases. The complexity of the exploratory stage may seem quite high, $O\left(N^4\right)$, since for each candidate connection we carry out several steps of the BPTT algorithm. Fortunately, experimental results prove that a few BPTT steps performed for each connection during the exploratory stage produces very good results, so the global cost remains low.

The experiments presented next concern univariate regression, but EBPTT is obviously not limited to such problems. Moreover, since the heuristic does not use any gradient information, we believe that it can be applied in combination with learning algorithms which are not gradient-based.

3. Experimental results

We applied EBPTT to RNN having a single input neuron, a single (linear) output neuron, a bias unit and a fully recurrent hidden layer composed of neurons with sigmoidal activation functions (Figure 1). For the sunspots dataset we tested RNN having 2 to 15 neurons in the hidden layer and for the Mackey-Glass dataset 2 to 7 neurons. We performed 20 experiments for every architecture, by randomly initializing the weights in $[-0.3, 0.3]$. We give here the best results we obtained for the two benchmarks and compare these results to several published ones.

In the following we employ the normalized mean squared error (NMSE) which is defined, for a time series $(x_t)_{t-t_1,\ldots,t_l}$, by

$$\frac{\sum_{t=t_1}^{t_l}(x_t-\hat{x}_t)^2}{\sum_{t=t_1}^{t_l}(x_t-\bar{x}_t)^2}=\frac{\sum_{t=t_1}^{t_l}(x_t-\hat{x}_t)^2}{l\sigma^2},$$

where $\hat{x}_t$ is the prediction given by the RNN and $\bar{x}$, σ^2 are the mean value and variance estimated from the available data.

3.1. Sunspots dataset

This dataset contains the yearly number of dark spots on the sun from 1700 to 1979. The time series has a pseudo-period of 10 to 11 years. Several models were evaluated for one step ahead predictions [20], [21], including feed-forward [22] and recurrent [23] neural networks. The training set corresponds to the period 1700-1920 and two test sets were defined, 1921-1955 (test1) and 1956-1979 (test2). Test2 is considered to be more difficult because it has a larger variance.

Table 1 compares the results obtained by various models applied to this benchmark. For every model we give the number of parameters and the NMSE. The Threshold AutoRegressive (TAR, [20]) model employs a threshold to switch between two autoregressive models. The MLP has a time window of size 12 in the input layer and starts with 8 hidden neurons [22]; a pruning algorithm reduces the number of hidden neurons to 3. The IIR MLP in [23] contains local feedbacks and delays and is obtained by an evolutionary algorithm. The Dynamical Recurrent Neural Networks (DRNN) are RNN having FIR connections [24]. DRNN1 has 2 hidden neurons fully connected by FIR connections of order 5, and DRNN2 has 5 hidden neurons fully connected by FIR connections of order 2. The order of these connections was found after several trials.

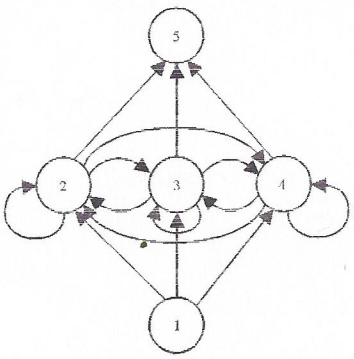

Figure 1: The recurrent architecture employed.

Model	Parameters	Learning	Test1	Test2
Carbon Copy	-	0.289	0.427	0.966
TAR	18	0.097	0.097	0.280
MLP	43	0.082	0.086	0.350
IIR MLP	23	0.101	0.097	0.436
DRNN1	30	0.105	0.091	0.273
DRNN2	45	0.111	0.093	0.246
RNN with BPTT	155	0.064	0.084	0.300
RNN with EBPTT	23	0.089	0.078	0.227

Table 1: NMSE obtained by various models on the sunspots time series.

We set to 20 the maximal value for the delays of the new connections, to 4 the maximal number of new connections and to 20 the number of BPTT steps performed for each candidate connection during the exploratory stage. We can notice the improvement upon BPTT (without the constructive stage). Moreover, the results produced with EBPTT are significantly better than those obtained by the other models, both for test1 and test2.

Connection added	Learning	Stop	Test1	Test2
-	0.102	0.195	0.110	0.318
1st	0.096	0.156	0.089	0.253
2nd	0.091	0.149	0.082	0.246
3rd	0.089	0.140	0.080	0.229
4th	0.089	0.139	0.078	0.227

Table 2: Evolution of the NMSE for the best RNN upon addition of new connections.

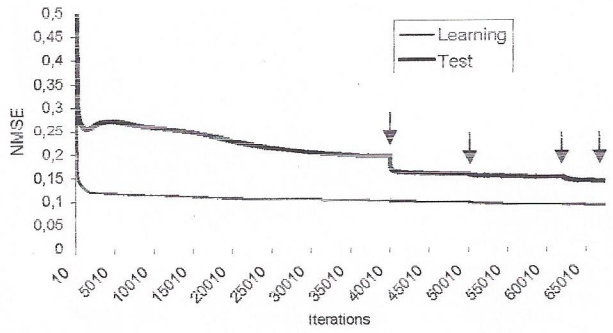

Figure 2: The evolution of the error for a single RNN. Every arrow corresponds to the addition of one connection.

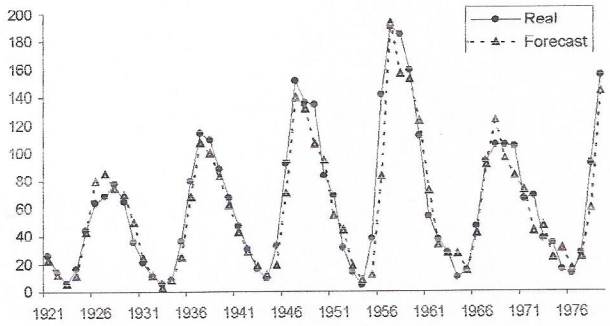

Figure 3: The predictions obtained with EBPTT on the sunspots test sets.

The number of parameters is also very low. The best results (see also Figure 3) were obtained for RNNs having only 3 neurons in the hidden layer. During our experiments we noticed that EBPTT always added 4 connections, with most of the delays between 3 and 10. The contribution of the new connections diminishes quickly as their number increases, as shown in Figure 2 and Table 2.

3.2. Mackey-Glass dataset

The Mackey-Glass time series [25] are generated by the following nonlinear model:

$$\frac{dx(t)}{dt} = -0.1x(t) + \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)}.$$

We consider here $\tau = 17$ (MG17), the value which is usually retained. The resulting time series (Figure 4) exhibits then a chaotic behavior. The data generated with $x(t) = 0.9$ for $0 \le t \le \tau$ is then sampled with a period of 6 (as in [7]). We use the first 500 values for the learning set and the next 100 values for the test set.

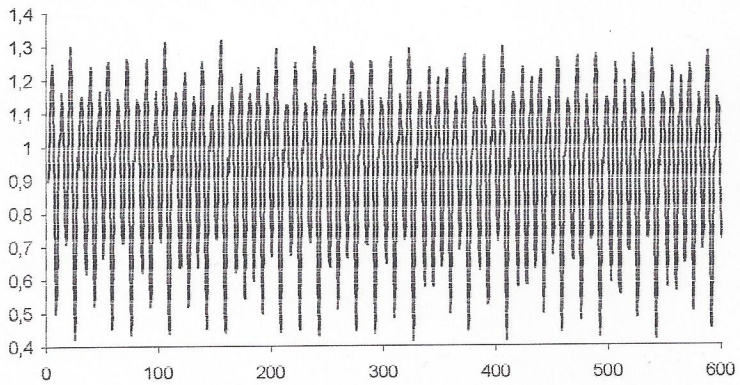

Figure 4: The Mackey-Glass time series for $\tau = 17$.

Table 3 compares the NMSE obtained on the test set by several models applied to this benchmark (see [6], [25], [26] for the first 3 models). The FIR MLP [6] has 15 neurons in the hidden layer; the FIR connections between the inputs and the hidden neurons have an order of 8, and those between the hidden neurons and the output an order of 2 (for a total of 196 parameters). In [27] a RNN having 5 neurons in a fully connected hidden layer is employed. The feed-forward network in [11] has a single input, 20 neurons in the hidden layer and one output neuron ; all the connections have delays, and the values of the delays are obtained by an algorithm which reminds back-propagation. The DRNN [24], [28] have FIR connections of order 4 between the input and the hidden layer, FIR connections of order 2 between the 4 to 7 hidden neurons, and simple connections to the output neuron (197 parameters).

We set to 34 the maximal value for the delays of the new connections, to 10 the maximal number of new connections and to 20 the number of BPTT steps performed for each candidate connection during the exploratory stage. EBPTT gives the best results for the Mackey-Glass dataset with $\tau = 17$. These results were obtained for RNN having 6 neurons in the hidden layer and 10 time-delayed connections, for a total of 65 parameters. During our experiments we noticed that EBPTT added 8.75 connections on the average, and their delays were distributed around the following values: 9, 20, 34. As for the sunspots data, only the first new connections produce a significant reduction of the NMSE (Figure 5).

We should mention that we performed similar experiments for $\tau = 30$ (MG30) and the EBPTT algorithm produced again the best results.

Model	NMSE on test set
Linear	0.269
RBF	1.07×10^{-2}
MLP	10^{-2}
FIR MLP	4.9×10^{-3}
RNN in [33]	3.1×10^{-3}
TDNN in [16]	8×10^{-4}
DRNN	4.7×10^{-3}
RNN with BPTT	2.35×10^{-4}
RNN with EBPTT	1.32×10^{-4}

Table 3: Results obtained by various models on the MG17 time series.

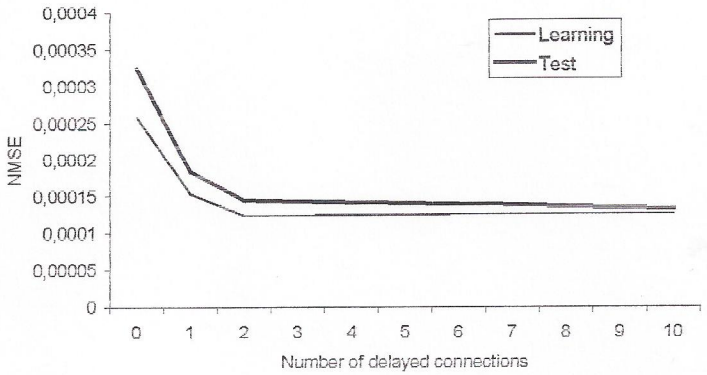

Figure 5: NMSE as a function of the number of connections added, for the best RNN found by EBPTT.

4. Conclusion

Adding time-delayed connections to recurrent neural networks helps gradient descent algorithms in learning medium or long-term dependencies. We opted here for a constructive approach, which starts with a RNN having no time-delayed connections and progressively adds a few such connections. We defined a heuristic for choosing the location and the (single) delay associated to a time-delayed connection. The heuristic is neither limited to regression problems, nor to recurrent networks. Note that this heuristic can be readily adapted to second order gradient-based algorithms.

The experimental results we obtained on two benchmark problems show that by adding only a small number of time-delayed connections one is able to produce networks having comparatively few parameters and good performance. Also, the number of learning steps performed with each candidate network during the exploratory stage is very low. This implies that the constructive part of the algorithm has only a minor impact on the global computational cost.

Acknowledgements

We gratefully acknowledge Yoshua Bengio for the interesting discussions related to the use of time-delayed connections in neural networks, discussions we had during the two months stay of one of the authors at the Université de Montréal.

References

1. Rumelhart, D.E., G.E. Hinton, and R.J. Williams, Learning Internal Representations by Error Propagation, in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, D.E. Rumelhart and J. McClelland, Editors, 1986, MIT Press: Cambridge, MA, p. 318-362.

2. Le Cun, Y., Modèles connexionnistes de l'apprentissage, *PhD Thesis*, 1987, Université Pierre et Marie Curie, Paris, France.
3. Cybenko, G., Approximation by Superpositions of a Sigmoidal Function, *Mathematics of Control, Signals and Systems*, 1989, 2: 304-314.
4. Hornik, K., M. Stinchcombe, and H. White, Multilayer Feedforward Networks Are Universal Approximators, *Neural Networks*, 1989, 2: 359-366.
5. Wan, E., Temporal Backpropagation for FIR Neural Networks, in *International Joint Conference on Neural Networks*, 1990, San Diego, USA, p. 575-580.
6. Back, A.D. and A.C. Tsoi, FIR and IIR Synapses, a New Neural Network Architecture for Time Series Modeling, *Neural Computation*, 1991, 3(3): 375-385.
7. Wan, E.A., Finite Impulse Response Neural Networks with Applications in Time Series Prediction, *PhD Thesis*, Department of Electrical Engineering, University of Stanford, 1993, 140 p.
8. Svarer, C., L.K. Hansen, and J. Larsen, On the Design and Evaluation of Tapped-Delay Neural Network Architectures, in *IEEE International Conference on Neural Networks*, 1993, San Francisco, p. 46-51.
9. Levin, A.U., T.K. Leen, and J.E. Moody, Fast Pruning Using Principal Components, in *Advances in Neural Information Processing Systems*, J. Cowan, G. Tesauro, and J. Alspector, Editors, 1994, Morgan Kaufmann: San Mateo, CA, p. 35-42.
10. Goutte, C., Extracting the Relevant Delays in Time Series Modelling, in *Neural Networks for Signal Processing VII*, 1997, Piscataway, New Jersey, USA, p. 92-101.
11. Duro, R.J. and J. Santos Reyes, Discrete-Time Backpropagation for Training Synaptic Delay-Based Artificial Neural Networks, *IEEE Transactions on Neural Networks*, 1999, 10(4): 779-789.
12. Seidl, D.R. and R.D. Lorenz, A Structure by which a Recurrent Neural Network Can Approximate a Nonlinear Dynamic System, in *International Joint Conference on Neural Networks*, 1991, p. 709-714.
13. Funahashi, K.I. and Y. Nakamura, Approximation of Dynamical Systems by Continuous Time Recurrent Neural Networks, *Neural Networks*, 1993, 6: 801-806.
14. Jin, L., N. Nikiforuk, and M.M. Gupta, Uniform Approximation of Nonlinear Dynamic Systems Using Dynamic Neural Networks, in *International Conference on Artificial Neural Networks*, 1995, Paris, France, p. 191-196.
15. Williams, R.J. and D. Zipser, A Learning Algorithm for Continually Running Fully Recurrent Neural Networks, *Neural Computation*, 1989, 1: 270-280.
16. Bengio, Y., P. Simard, and P. Frasconi, Learning Long-Term Dependencies with Gradient Descent is Difficult, *IEEE Transactions on Neural Networks*, 1994, 5(2): 157-166.
17. El Hihi, S. and Y. Bengio, Hierarchical Recurrent Neural Networks for Long-Term Dependencies, in *Advances in Neural Information Processing Systems*, M. Mozer, D.S. Touretzky, and M. Perrone, Editors, 1996, MIT Press: Cambridge, MA, p. 493-499.
18. Lin, T., et al., Learning Long-Term Dependencies in NARX Recurrent Neural Networks, *IEEE Transactions on Neural Networks*, 1996, 7(6): p. 1329-1338.
19. Guignot, J. and P. Gallinari, Recurrent Neural Networks with Delays, in *International Conference on Artificial Neural Networks*, 1994, Sorrento, Italy, p. 389-392.
20. Tong, H. and K.S. Lim, Threshold Autoregression, Limit Cycles and Cyclical Data, *Journal of the Royal Statistical Society*, 1980, B42: 245-292.
21. Priestley, M.B., *Spectral Analysis and Time Series*, 1981: Academic Press.
22. Weigend, A.S., B.A. Huberman, and D.E. Rumelhart, Predicting the Future: A Connectionist Approach, *International Journal of Neural Systems*, 1990, 1(3): 193-209.
23. McDonnell, J.R. and D. Waagen, Evolving Recurrent Perceptrons for Time Series Modeling, *IEEE Transactions on Neural Networks*, 1994, 5(1): 24-38.
24. Aussem, A., Théorie et applications des réseaux de neurones récurrents et dynamiques à la prédiction, à la modélisation et au contrôle adaptatif des processus dynamiques, *PhD Thesis*, U.F.R. de Mathématiques et Informatique, Université René Descartes, Paris, France, 1995, 138 p.
25. Casdagli, M., Nonlinear Prediction of Chaotic Time Series, *Physica*, 1989, 35D: p. 335-356.
26. Svarer, C., et al. Designer Networks for Time Series Processing, in *Neural Networks for Signal Processing III*, 1993, p. 78-87.
27. Logar, A.M., E.M. Corwin, and W.J.B. Oldham. A Comparison of Recurrent Neural Network Learning Algorithms, in *IEEE International Conference on Neural Networks*, 1993, San Francisco, p. 1129-1134.
28. Aussem, A., Nonlinear Modeling of Chaotic Processes with Dynamical Recurrent Neural Networks, in *NEURal networks and their APplications*, Marseille, France, 1998, p. 425-433.