ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 293-298

An Algorithm for the Addition of Time-Delayed Connections to Recurrent Neural Networks

Romuald Boné, Michel Crucianu, 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, asselin}@univ-tours.fr

Abstract: Recurrent neural networks possess interesting universal approximation capabilities, making them good candidates for time series modeling. Unfortunately, long term dependencies are difficult to learn if gradient descent algorithms are employed. We support the view that it is easier for these algorithms to find good solutions if one includes connections with time delays in the recurrent networks. The algorithm we present here allows one to choose the right locations and delays for such connections. As we show on two benchmark problems, this algorithm produces very good results while keeping the total number of connections in the recurrent network to a minimum.

1. Introduction

Time series processing has important applications in various domains such as medicine, ecology, meteorology, industrial control or finance. The most common approach to modeling 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 (regression) or the class membership of the input sequence (classification). In univariate regression, for instance, one is searching for the function f which gives the best estimate of the future value of the time series according to $\hat{x}_t = f\left(x_{t-1}, x_{t-2}, \ldots x_{t-p}\right)$, p being the size of the time window.

Multilayer perceptrons (MLP, [1]) are well adapted to this approach: one is simply using a network having an input layer of size p, several 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 show that very general nonlinear autoregressive functions $f_{\mathbf{w}}$ can be obtained; these functions depend on the weights in the network. Replacing simple connections by finite impulse response (FIR) connections produces composite nonlinear autoregressive models which give better results on several reference benchmarks [2].

However, this approach is unable to model well the behavior of many dynamical systems and is often limited by the presence of a time window of fixed size. Indeed, if the time window is too narrow, important cues may fall aside. If the window is too wide, useless inputs may act as noise. Ideally, the size of the window should adapt to the context, but it is difficult for an MLP to implement this feature.

ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 293-298

Recurrent neural networks (RNN) possess an internal memory and do no longer need a time window to take into account the past values of the time series. RNNs prove to be significantly more powerful than feed-forward networks, for regression [3] or classification [4] problems. Unfortunately, the gradient descent algorithms which are commonly used for training RNNs [1], [5] have several limitations, the most important one being the difficulty of dealing with long-term dependencies in the time series [6]. Adding connections with time delays to the RNN [7], [8] often allows gradient descent algorithms to find better solutions in these cases. Indeed, by acting as a linear link between two distant moments, such a connection has beneficial effects on the expression of the gradient.

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 adding FIR connections, each encompassing a whole range of delays, one obtains oversized networks which are slow to train and have poor generalization abilities. Various regularization techniques are then employed in order to improve generalization [9], [10], and this further increases the computational cost.

2. A constructive algorithm for time-delayed connections

We opted here for the alternative, constructive approach: start with a RNN having no time-delayed connections and progressively add a few such connections. We choose the location and the (single) delay associated to a time-delayed connection by evaluating the relevance of the potential candidates. The resulting algorithm allows us to better account for long-term dependencies by adapting the architecture of the RNN to the problem we must solve.

Let us consider Back Propagation Through Time (BPTT) [1]. When applying BPTT on the training set between t_1 and t_l, we obtain the following expression for the variation of one weight $w_{ij}^{(k)}$ of delay k:

$$\Delta w_{ij}^{(k)} = -\eta \sum_{\tau=t_1+k}^{t_l-1} \frac{\partial E(t_1,t_l)}{\partial w_{ij}^{(k)}(\tau)},$$

$E(t_1,t_l)$ being the mean quadratic error and $w_{ij}^{(k)}(\tau)$ the copy of $w_{ij}^{(k)}$ for $t=\tau$ in the unfolded network BPTT employs [1]. If we note by $s_j(t)$ the output of neuron j at time t, we may write

$$\frac{\partial E(t_1,t_l)}{\partial w_{ij}^{(k)}(\tau)} = \frac{\partial E(t_1,t_l)}{\partial net_i(\tau)} s_j(\tau-k) \qquad \tau \in [t_1+k; t_l-1].$$

The connection $w_{ij}^{(k)}$ is only useful in capturing long-term dependencies if it has a significant contribution to the computation of the gradient, i.e. $\partial E(t_1,t_l) \big/ \partial w_{ij}^{(k)}(\tau)$ is significantly non zero for many iterations of the learning algorithm.

ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 293-298

Our algorithm, Constructive Back Propagation Through Time (CBPTT), records during several BPTT steps the correlation between the values of $\partial E(t_1, t_l)/\partial net_i(\tau)$ and $s_j(\tau - k)$ for $\tau \in [t_1 + k; t_l - 1]$. The absolute value of this correlation defines the relevance factor for every candidate connection $w_{ij}^{(k)}$. The connection having the highest relevance factor is then added to the RNN (its weight is initialized to 0) and learning continues. This process usually stops when a new connection has no further positive effect on the performance of the network. Note that the storage complexity and the time complexity of CBPTT is the same as for BPTT.

3. Experimental results

The experimental results we present here concern univariate regression, but CBPTT is not limited to such problems. We applied CBPTT 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. 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. 20 experiments were performed for every architecture, by randomly initializing the weights in $[-0.3, 0.3]$. We show the best results we obtained for the two benchmarks and compare these results to published ones.

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 [11], [12], including feed-forward [13] and recurrent [14] 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 normalized mean squared error (NMSE). The Threshold AutoRegressive (TAR, [11]) 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 [13]; a pruning algorithm reduces the number of hidden neurons to 3. The IIR MLP in [14] contains local feedbacks and delays and is obtained by an evolutionary algorithm.

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
RNN with BPTT	155	0.064	0.084	0.300
RNN with CBPTT	15	0.098	0.092	0.251

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

ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 293-298

The results produced with CBPTT are in line with those of the other models on test1, but are significantly better on test2. The number of parameters is also very low. The best results (see also Figure 1 below) were obtained for RNNs having only 2 neurons in the hidden layer. During our experiments we noticed that CBPTT added at most 4 connections, with delays between 6 and 17.

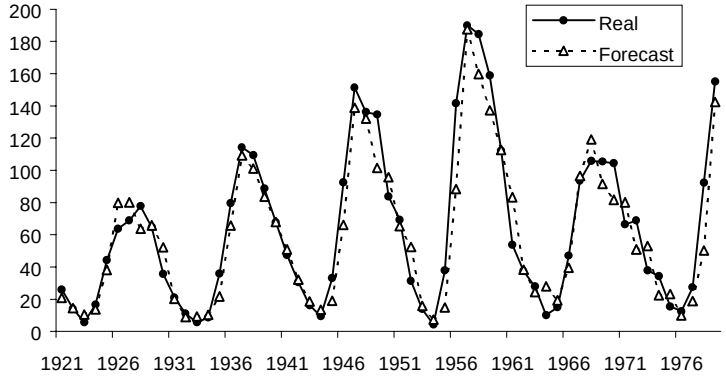

Figure 1: The predictions obtained with CBPTT on the sunspots test sets.

3.2. Mackey-Glass dataset

The Mackey-Glass time series [16] 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 time series 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. We use the first 500 values for the learning set and the next 100 values for the test set.

Table 2 compares the results (NMSE) obtained on the test set by several models applied to this benchmark (see [2], [15], [16] for the first 7 models). The FIR MLP [2] 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 [17] a RNN having 5 neurons in a fully connected hidden layer is employed. The feed-forward network in [18] 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 is similar to back-propagation.

CBPTT gives the best results for the Mackey-Glass dataset with $\tau = 17$. These results were obtained for RNN having 7 neurons in the hidden layer and 4 time-delayed connections, for a total of 81 parameters. During our experiments we noticed

ESANN'2000 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2000, D-Facto public., ISBN 2-930307-00-5, pp. 293-298

that CBPTT added at most 4 connections, and their delays were distributed around the following values: 5, 13, 19. Note that we performed similar experiments for $\tau = 30$ (MG30) and CBPTT produced again the best results.

Model	Results on test set
Linear	0.269
Polynomial	$1.12 \cdot 10^{-2}$
Rational	$7.24 \cdot 10^{-2}$
Local approach 1	$3.31 \cdot 10^{-2}$
Local approach 2	$1.29 \cdot 10^{-2}$
RBF	$1.07 \cdot 10^{-2}$
MLP	10^{-2}
FIR MLP	$4.9 \cdot 10^{-3}$
RNN in [18]	$3.1 \cdot 10^{-3}$
TDNN in [19]	$8 \cdot 10^{-4}$
RNN with BPTT	$2.35 \cdot 10^{-4}$
RNN with CBPTT	$1.4 \cdot 10^{-4}$

Table 2: Results obtained by various models on the MG17 time series.

4. Conclusion

Adding time-delayed connections to recurrent neural networks helps gradient descent algorithms in learning medium or long-term dependencies. However, by systematically adding finite impulse response connections, one obtains oversized networks which are slow to train and need difficult to control regularization techniques in order to improve generalization.

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 resulting algorithm has the same computational complexity as the well known back-propagation through time.

The experimental results we obtained on two benchmark problems show that by adding only a few time-delayed connections one is able to produce networks having comparatively few parameters and good performance.

Our effort is now directed towards adapting this heuristic to second-order gradient based algorithms and defining alternative relevance factors.

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. Wan, E.A., Finite Impulse Response Neural Networks with Applications in Time Series Prediction, *PhD Thesis*, 1993, Stanford University, 140 p.
3. 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, Vol. 2, p. 709-714.
4. Santini, S. and A. Del Bimbo, Recurrent neural networks can be trained to be maximum a posteriori probability classifiers, *Neural Networks*, 1995, **8**: 25-29.
5. Williams, R.J. and D. Zipser, A Learning Algorithm for Continually Running Fully Recurrent Neural Networks, *Neural Computation*, 1989, **1**: 270-280.
6. 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.
7. 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.
8. Lin, T., et al., Learning Long-Term Dependencies in NARX Recurrent Neural Networks, *IEEE Transactions on Neural Networks*, 1996, **7**(6): 1329-1337.
9. Guignot, J. and P. Gallinari, Recurrent Neural Networks with Delays, in *International Conference on Artificial Neural Networks*, 1994, Sorrento, Italy.
10. Giles, C.L. and C.W. Omlin, Pruning Recurrent Neural Networks for Improved Generalization Performance, *IEEE Transactions on Neural Networks*, 1994, **5**(5): 848-851.
11. Tong, H. and K.S. Lim, Threshold Autoregression, Limit Cycles and Cyclical Data, *Journal of the Royal Statistical Society*, 1980, **B42**: p. 245-292.
12. Priestley, M.B., *Spectral Analysis and Time Series*, 1981: Academic Press.
13. 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.
14. McDonnell, J.R. and D. Waagen, Evolving Recurrent Perceptrons for Time Series Modeling, *IEEE Transactions on Neural Networks*, 1994, **5**(1): 24-38.
15. Casdagli, M., Nonlinear Prediction of Chaotic Time Series, *Physica*, 1989. **35D**: 335-356.
16. Svarer, C., et al. Designer Networks for Time Series Processing, in *Neural Networks for Signal Processing* III, 1993.
17. 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: IEEE.
18. 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.