Bi-causal Recurrent Cascade Correlation

A. Micheli D. Sona A. Sperduti
Dipartimento di Informatica, Università di Pisa
Corso Italia, 40, 56125 Pisa, Italy
e-mail: {micheli,sona,perso}@di.unipi.it

Abstract

Recurrent neural networks fail to deal with prediction tasks which do not satisfy the causality assumption. We propose to exploit bi-causality to extend the Recurrent Cascade Correlation model in order to deal with contextual prediction tasks. Preliminary results on artificial data show the ability of the model to preserve the prediction capability of Recurrent Cascade Correlation on strict causal tasks, while extending this capability also to prediction tasks involving the future.

1 Introduction

Basically, all the recurrent neural network models proposed in literature are based on a causality assumption, i.e., the output of the network at time t_0 only depends on input at times $t \leq t_0$. Because of that, several prediction tasks involving sequences require processing of information from both the past and the future. The prediction of the secondary structure of proteins, as well as language understanding, are examples of these tasks.

Typical approaches to these tasks involve feed-forward networks that looks at the input through a fixed window of predefined size [QS88]. An extension of this approach is given by Time Delay networks [WHH$^+$89], for which however predefined window sizes must be specified as well.

Some authors (Baldi et al. [BBF$^+$99]) suggested to solve the fixed size window problem by factoring the internal state of a recurrent neural network into a forward state and a backward state. These components propagate information from the past and from the future, respectively, and they are combined to produce the output.

In this paper, we formalize the concept of *bi-causality*, which is at the basis of the approach reported above, and we propose an instance of bi-causality suitable for implementation into Recurrent Cascade Correlation [Fah91]. These networks are able to automatically devise a near-optimal internal state size according to the training data, thus determining automatically the "right" number of hidden units.

Preliminary results on artificially generated data show the ability of the proposed model (Bi-causal Recurrent Cascade Correlation) to deal with dependencies on the future. Moreover, the experimental results show that this ability does not impair the model performance on strict causal prediction tasks.

In the next section we discuss bi-causality, while in Section 3 we define the Bi-causal Recurrent Cascade Correlation model, showing how to compute the gradient. Experimental results are presented and briefly discussed in Section 4, where we also show that Recurrent Cascade Correlation is unable to deal with tasks solved by the proposed model. Conclusions are drawn in Section 5.

2 Causality vs Bi-causality

Recurrent neural networks possess, in principle, the ability to memorize past information to perform complex sequential mappings. This ability is based on the assumption that a propriety called *causality* holds, i.e., a system is *causal* if the output at time t_0 depends only on inputs at times $t \leq t_0$. Formally speaking, a causal system is described by the following equations:

$$\begin{cases} \boldsymbol{x}(t) &= \tau(\boldsymbol{x}(t-1), \boldsymbol{u}(t), t) \\ \boldsymbol{y}(t) &= g(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \end{cases} \tag{1}$$

where $\boldsymbol{u}(t)$ is the input at time t, $\boldsymbol{x}(t)$ is the internal state of the system at time t, and $\boldsymbol{y}(t)$ is the output of the system at time t. Note that the above equations describe a non-stationary system. Typically, the system is assumed to be stationary as well, i.e., neither the state transition function τ nor the output function g depend on t.

In several applications concerning sequence processing, however, causality is not sufficient. For example, all the tasks that require contextual information involving both the past and the future violate the causality assumption. There are many ways to access contextual information in a sequence of elements [QS88, WHH$^+$89], however, for all of them predefined window sizes must be specified. A solution to this problem is to resort to recurrent networks, which however assume causality.

Here we suggest a specific extension of the causality concept, *bi-causality*, which introduces also a dependency on the future in a form amenable to its application in Recurrent Cascade Correlation networks. In general, we could define the bi-causality as a double and simultaneous causal dependency over the past and the future. For example, Baldi et al. [BBF$^+$99] propose a factorization of the internal state in two parts, $\boldsymbol{x}(t) = [\boldsymbol{x}_1(t), \boldsymbol{x}_2(t)]^t$, where $\boldsymbol{x}_1 \in I\!R^m$ and $\boldsymbol{x}_2 \in I\!R^n$ keep information about the past (left-to-right direction) and the future (right-to-left direction), respectively. The state transition function of the system is then defined as:

$$\boldsymbol{x}(t) \equiv \begin{bmatrix} \boldsymbol{x}_1(t) \\ \boldsymbol{x}_2(t) \end{bmatrix} = \begin{bmatrix} \tau_1(\boldsymbol{x}_1(t-1), \boldsymbol{u}(t), t) \\ \tau_2(\boldsymbol{x}_2(t+1), \boldsymbol{u}(t), t) \end{bmatrix}. \quad (2)$$

The above idea as been implemented by Baldi et al. [BBF$^+$99] through a Bidirectional Recurrent Neural Network (BRNN) which can be understood as composed of three subnetworks: one for computing $\boldsymbol{x}_1$, one for $\boldsymbol{x}_2$, and finally one subnetwork which combines all the information for producing the output.

The maintenance of such a huge network can be quite difficult, as well as the decision about how many hidden units should be used for the computation of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. For this reason we propose a different realization of bi-causality which is amenable to Recurrent Cascade Correlation networks, thus leaving to the learning algorithm the decision on how many hidden units are needed for the specific task at hand.

The main idea about our new formulation comes from the observation that in Recurrent Cascade Correlation networks hidden units are frozen one by one as new units are added. Notice that, each time a unit is frozen, the training procedure has already observed the whole sequence, thus the portion of the state associated to that unit encodes "knowledge" of the whole sequence. This knowledge can be accessed without problems by new hidden units. One possible formulation of the state transition function which considers the above observation on component i at time t is given by the following equation:

$$x_i(t) = \tau_i(x_i(t-1), x_{i-1}(t-1), x_{i-1}(t+1), x_{i-2}(t-1), x_{i-2}(t+1), \ldots, x_1(t-1), x_1(t+1), u(t), t) \quad (3)$$

where the initial state is used when needed, both for $x_k(t-1)$ and $x_k(t+1)$. Notice that information about the future can be accessed by component i since the components up to $i-1$ are computed by frozen hidden units in the Recurrent Cascade Correlation network.

A graphical model of the above equations for $i \in \{1, 2, 3\}$ is shown in Figure 1. Note that each unit takes input both from the past and from the future of preceding units. Concerning the future, in Figure 1 we have highlighted the dependency of $x_i(t_0)$ on input labels at times up to $t = t_0 + i - 1$, so to show that the window size on the future is directly proportional to the number of hidden units.

3 The Bi-causal Recurrent Cascade Correlation Network

Referring to eq. 3, and assuming stationarity, i.e., independence of $\tau_i(\cdot)$ on t, the output of the kth hidden unit (corresponding to x_k), in our Bi-causal Recurrent Cascade Correlation Network (BRCC), can be computed as

$$x_k(t) = f(\sum_{i=0}^{n} w_{ki} l_i(t) + \sum_{v=1}^{k} \hat{w}_{kv} x_v(t-1) + \sum_{q=1}^{k-1} \tilde{w}_{kq} x_q(t+1)) \quad (4)$$

where $f(\cdot)$ is a sigmoidal (or radial) function, $\boldsymbol{l}(t) \in I\!R^{n+1}$ is the input label at time t (with $l_0(t) = 1$ for each t), $\boldsymbol{w}_k \in I\!R^{n+1}$ is the weight vector associated with the input label (with w_{k0} as threshold), $\hat{\boldsymbol{w}}_k \in I\!R^k$ is the weight vector associated with the output of the hidden units at time $t-1$ (where units up to $k-1$

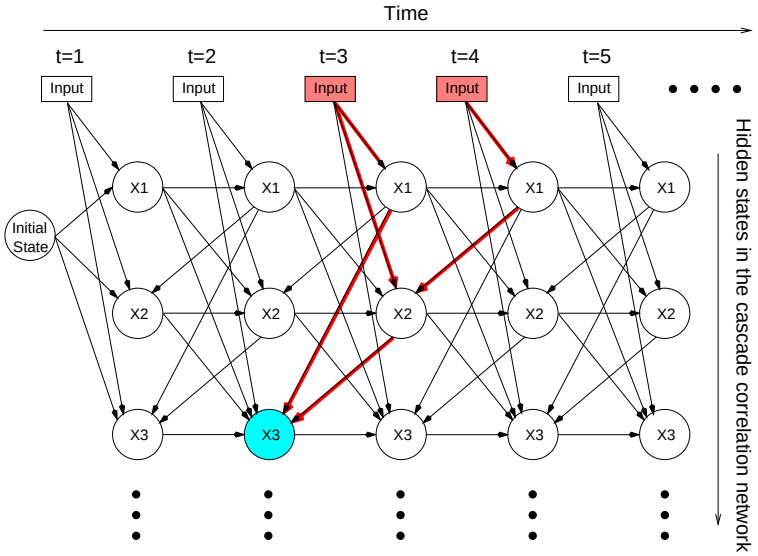


Figure 1: Graphical model (for $i \in \{1, 2, 3\}$) of our bi-causal formulation of the state transition function (eq. 3). We have emphasized links which convey information about the future to x_3 at time $t = 2$.

are frozen), and $\tilde{\boldsymbol{w}}_k \in I\!R^{k-1}$ is the weight vector associated with the output of the frozen hidden units at time $t + 1$.

The output of the network (with k inserted hidden units and assuming a single output unit) is then computed as

$$o(t) = g(\boldsymbol{x}(t), \boldsymbol{u}(t)) = f(\boldsymbol{A}^t \boldsymbol{x}(t) + \boldsymbol{B}^t \boldsymbol{u}(t) + \theta), \tag{5}$$

where $\boldsymbol{A} \in I\!R^n$ and $\boldsymbol{B} \in I\!R^k$ are the output weight vectors for the hidden state and the input, respectively, and θ is the output threshold. In this paper, since we are interested in predicting continuous values, the output function is set to be linear. Moreover, in our experiments we have removed the direct connections from the input to the output, i.e.,

$$o(t) = \boldsymbol{A}^t \boldsymbol{x}(t) + \theta. \tag{6}$$

Learning is performed as in standard Cascade Correlation by interleaving the minimization of the total error function (LMS) and the maximization of the correlation of the new inserted hidden unit with the residual error. The main difference with respect to standard Cascade Correlation is in the calculation of the derivatives. According to equation (4), the derivatives of $x_k(t)$ with respect to the weights are computed as

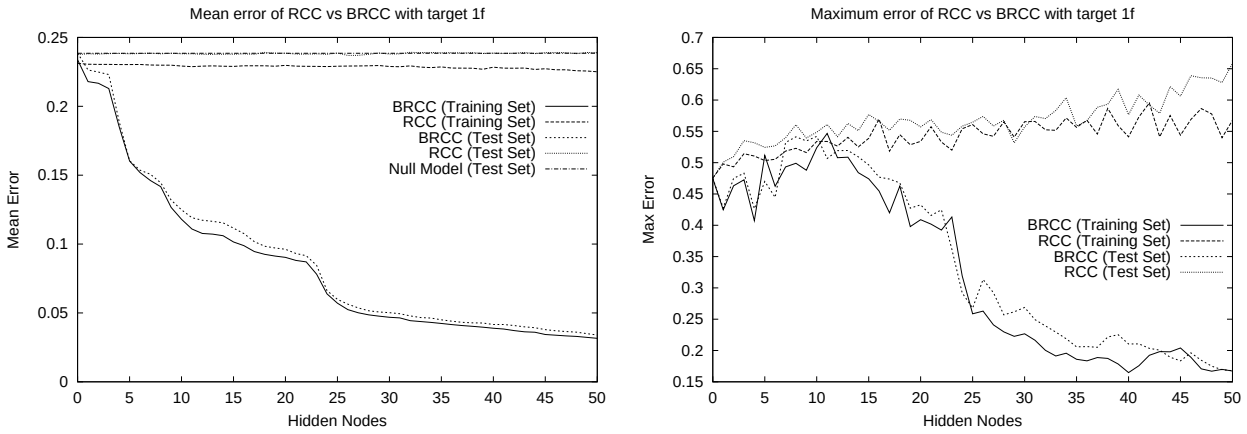
$$\frac{\partial x_k(t)}{\partial w_{ki}} = f'(l_i(t) + \hat{w}_{kk} \frac{\partial x_k(t-1)}{\partial w_{ki}}), \qquad i = 0, \ldots, n \tag{7}$$

$$\frac{\partial x_k(t)}{\partial \hat{w}_{kv}} = f'(x_v(t-1) + \hat{w}_{kk} \frac{\partial x_k(t-1)}{\partial \hat{w}_{kv}}), \quad v = 1, \ldots, k \tag{8}$$

$$\frac{\partial x_k(t)}{\partial \tilde{w}_{kq}} = f'(x_q(t+1) + \hat{w}_{kk} \frac{\partial x_k(t-1)}{\partial \tilde{w}_{kq}}), \quad q = 1, \ldots, k-1 \tag{9}$$

where f' is the derivative of $f(\cdot)$. Note that equations (7) and (8) are the same used for standard Recurrent Cascade Correlation (RCC), while equation (9) is added so to include also future information from frozen units. The above equations are recurrent and can be computed by observing that for the first element of each sequence $\frac{\partial x_k(t)}{\partial w_{ki}} = l_i f'$, and all the remaining derivatives are null. Consequently, we only need to store the output values of the unit and its derivatives for each component of a sequence.

Learning for the output weights proceeds as in the standard Cascade Correlation algorithm.



(a) (b)

Figure 2: Mean (a) and maximum (b) error of BRCC and RCC. Note that the mean error of RCC is near the mean error of the null model, and the maximum error diverges.

4 Experimental Results

In this section, we report some preliminary experiments on artificial data sets in order to evaluate the ability of BRCC in learning contextual mappings. The aim of our experiments is to show that while RCC is unable to learn a contextual mapping, BRCC can do it. Furthermore, this ability of the BRCC does not impair the prediction ability of the model under strict causality conditions.

For our experiments, we have considered regression problems. Different sets of randomly generated artificial sequences have been produced. The training sets are composed of 200 sequences while the test sets are composed of 100 sequences. The sequences, of length between 5 and 20, are composed of symbols in the alphabet $\mathcal{A} = \{\mathtt{a}, \ldots, \mathtt{j}\}$. Each symbol is selected according to a uniform distribution over the alphabet and it is coded as a 10-bit string, with one specific bit turned on (+1) and all others turned off (−1). Moreover, in order to define the target, a function $v : \mathcal{A} \to \{0, 0.1, \ldots, 0.9\}$ is defined (i.e., $v(\mathtt{a}) = 0, \ldots, v(\mathtt{j}) = 0.9$). Different prediction tasks were obtained by defining different target functions for each element of a sequence.

The first target function, strictly dependent on the next position in the sequence, is defined as in the following

$$target_{1f}(t) = v(s_i(t+1)), \tag{10}$$

where s_i is the i-th sequence, and $s_i(t+1)$ returns the sequence element in position $t+1$. For comparison with RCC we have also used the following causal target function (which depends only on the past element):

$$target_{1p}(t) = v(s_i(t-1)). \tag{11}$$

Other target functions involving the average on a window of size 4 have been used:

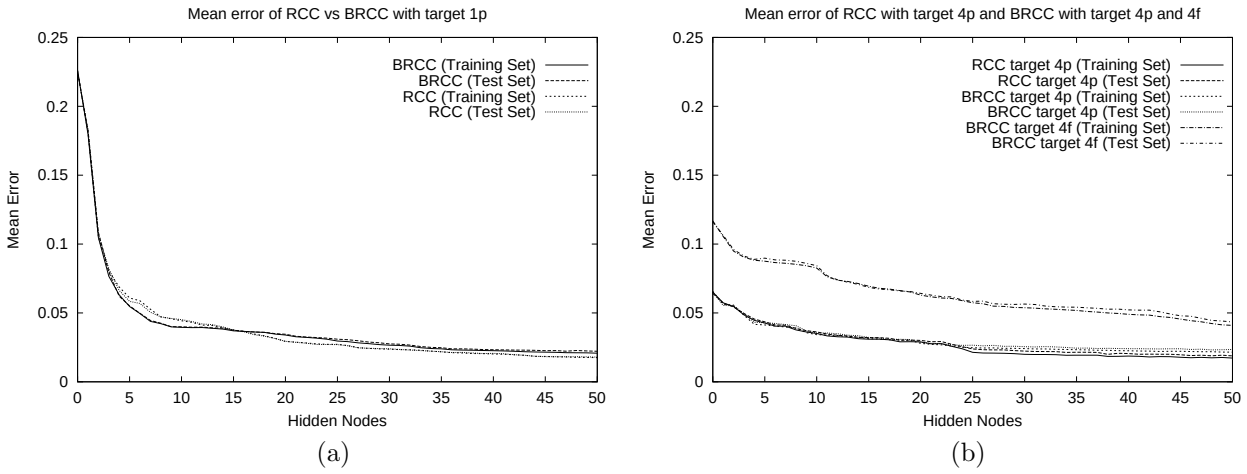
$$target_{4f}(t) = \frac{v(s_i(t)) + v(s_i(t+1)) + v(s_i(t+2)) + v(s_i(t+3))}{4}, \tag{12}$$

$$target_{4p}(t) = \frac{v(s_i(t)) + v(s_i(t-1)) + v(s_i(t-2)) + v(s_i(t-3))}{4}. \tag{13}$$

Finally, we have defined a moving average target over the future:

$$target_{ma}(t) = \frac{target_{ma}(t+1) + v(s_i(t))}{2}. \tag{14}$$

We performed several training trials with all the above defined target functions. Here we report examples of specific trials which are representative of the behavior of BRCC and RCC.



,

Figure 3: Mean error for RCC and BRCC over: (a) $target_{1p}$; (b) $target_{4p}$ and $target_{4f}$.

An example of the results obtained by BRCC and RCC over $target_{1f}$ are given in Figure 2(a). The performance of a theoretical null model[1] for the test set is shown as well. Note that, as expected, the RCC is not able to improve over the null model. The difficulty of RCC to deal with the prediction task is also evident from the increase in the maximum error corresponding to the increase of the number of hidden units into the network (Figure 2(b)). On the contrary, BRCC is able to decrease the maximum error along with the increase in the number of hidden units.

As shown in Figure 3(a), when considering $target_{1p}$ (i.e., the causality assumption holds) the results obtained by BRCC are comparable with those obtained by RCC. This shows that the BRCC's ability to use contextual information does not impair the performance of the model under strict causal conditions. A confirmation of this behavior is given when experimenting with $target_{4p}$ (see Figure 3(b)).

Finally, BRCC seems to be able to cope well with longer dependencies in the future, as encoded in $target_{4f}$ (see Figure 3(b)), as well as with the moving average over the future, i.e., $target_{ma}$ (see Figure 4).

Note that in all the experiments we let the algorithms to insert many more hidden units than necessary for solving the regression problems, however, it can be observed that no overfitting was observed.

[1]The null model is obtained by computing the expected value for the target over the training set.

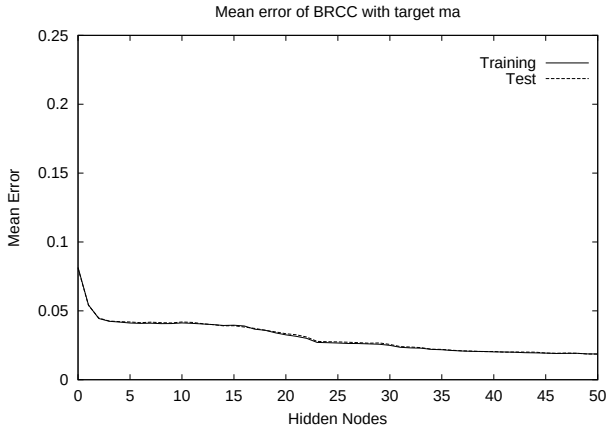


Figure 4: Mean error for BRCC on $target_{ma}$.

5 Conclusions

In this paper we have proposed a specific extension of the causality concept, *bi-causality*, which introduces also a dependency on the future in a form amenable to its application in Recurrent Cascade Correlation networks. Using this extension, we have proposed the Bi-causal Recurrent Cascade Correlation algorithm, able to perform contextual analysis of sequences.

Preliminary results on a range of different regression tasks, including either past or future dependencies, showed that the proposed model does not reduce the efficiency over past knowledge representation, while being able to capture the dependency of the target on the future.

The proposed model is strictly correlated with the one proposed by Baldi et al. [BBF$^+$99] (BRNN), however, the Bi-causal Recurrent Cascade Correlation model holds the advantage of building incrementally the network according to the computational needs. Specifically, the window size over the future is proportional to the number of hidden units inserted by the algorithm. Moreover, our approach is orthogonal to the one used in BRNN and could be combined with it. In conclusion, from the preliminary results, it seems that the BRCC should be preferred to RCC when no information on the causality assumption validity is known.

We plan to apply our model to prediction tasks in DNA analysis. Moreover, the bi-causality concept will be extended for dealing with structured domains [SS97, FGS98].

References

[BBF$^+$99] P. Baldi, S. Brunak, P. Frasconi, G. Pollastri, and G. Soda. Bidirectional dynamics for protein secondary structure prediction. In C.L. Giles and R. Sun, editors, *IJCAI'99: Workshop on Neural, Symbolic, and Reinforcement Methods for Sequence Learning*, 1999.

[Fah91] S.E. Fahlman. The recurrent cascade-correlation architecture. In R.P. Lippmann, J.E. Moody, and D.S. Touretzky, editors, *Advances in Neural Information Processing Systems 3*, pages 190–196, San Mateo, CA, 1991. Morgan Kaufmann Publishers.

[FGS98] P. Frasconi, M. Gori, and A. Sperduti. A framework for adaptive data structures processing. *IEEE Transactions on Neural Networks*, 9(5):768–786, 1998.

[QS88] N. Qian and T. J. Sejnowski. Predicting the secondary structure of globular proteins using neural network models. *Journal of Molecular Biology*, 202:865–884, 1988.

[SS97] A. Sperduti and A. Starita. Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks*, 8(3):714–735, 1997.

[WHH$^+$89] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang. Phoneme recognition using time–delay neural networks. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(3):328–339, 1989.