Recursive Cascade Correlation for Contextual Processing of Structured Data

Alessio Micheli, Diego Sona, Alessandro Sperduti
Dipartimento di Informatica, Università di Pisa
Corso Italia 40, 56125, Pisa, Italy
E-mail: {micheli,sona,perso}@di.unipi.it

Abstract - We propose an extension of Recursive Cascade Correlation (RCC) for structured domains which is able to partially remove the causality assumption. In fact, the proposed model, i.e. Contextual Recursive Cascade Correlation, is able to exploit contextual information stored in frozen units. Experimental results, obtained on the prediction of the boiling point of alkanes, show the superiority of the proposed approach versus RCC.

I. INTRODUCTION

Basically, almost all the recurrent neural network models proposed in literature are based on the causality assumption, i.e., the output of the network at time t_0 only depends on input at times $t \leq t_0$. Nevertheless, several prediction tasks involving sequences require processing of information from both the past and the future. Typical approaches to these tasks involve feed-forward neural networks that look at the input through a fixed window of predefined size [6].

Some authors suggested to solve the fixed size window problem by specific models which compute the output by combining information propagated both from the past and the future. This is performed spanning the sequence in the two directions. For example, in the recurrent model proposed in [1], the internal state is factorized into a forward state and a backward state. In particular the devised *bidirectional recurrent neural network* is composed of three sub-networks: one for computing the "past" information, one for computing the "future" information, and finally one sub-network which combines all the information to produce the output. A related approach has been proposed in [8].

A different approach has been introduced in [5] where the proposed model is a variant of the basic recurrent cascade correlation (RCC) [4], referred as *bi-causal recurrent cascade correlation* (BRCC). Actually, when training an RCC, hidden units are frozen one by one as new units are added. Since weights of frozen units are not allowed to change, it is possible to use the state information of the frozen units to also analyze an internal representation of the "future" inputs. When training a new hidden unit the information stored in frozen units can be accessed. In this way, when processing a sequence s at a time t, it is possible to use the stored activations for all the following subsequences of s,

$$s_{[0,1]}, s_{[0,2,]}, \ldots, s_{[0,t-1]}, s_{[0,t]}, s_{[0,t+1]}, \ldots, s_{[0,t_s]}$$

where $s_{[i,j]}$ is the subsequence of s in the interval $[i,j]$ and t_s is the length of the sequence s.

The recursive neural network model [7], a generalization of the recurrent model able to deal with structured information (trees, DOAGs, etc), inherits a causality assumption defined on structured data. In the framework of structure processing, the model is *causal* if the output for a given vertex of a directed graph depends only on the current vertex and the vertexes descending from it. This assumption allows to use internal states to memorize information about substructures.

As in the case of sequences, causality is not sufficient when the task requires complete contextual information, or, more in general, when there is no knowledge supporting the causality assumption. For instance, non-causal models can be useful when dealing with structured data where the meaning of sub-structures depends from the context in which they are found. The challenge is therefore to study the possibility to process structures by a recursive neural network model, relaxing the causality assumption.

In the following we describe a *contextual recursive cascade correlation* for structures (CRCC), based on an extension of BRCC [5], able to perform contextual processing of structured data (sequences in the simplest form).

II. STRUCTURED DOMAINS AND CONTEXTUAL RECURSIVE NEURAL MODEL

In this paper we assume that instances in the learning domain are DPAGs (directed positional acyclic graphs). A DPAG is a DAG $\mathcal{D}$ with vertex set $\mathrm{vert}(\mathcal{D})$ and edge set $\mathrm{edg}(\mathcal{D})$, where we assume that for each vertex $v \in \mathrm{vert}(\mathcal{D})$, a bijection $P_v : \mathrm{edg}(v) \rightarrow I\!N$ is defined on the edges entering and leaving from v, i.e. all edges are numbered with a positional index. We shall require the DPAGs to possess a supersource[1], i.e. a vertex $s \in \mathrm{vert}(\mathcal{D})$ such that every vertex in $\mathrm{vert}(\mathcal{D})$ can be reached by a directed path starting from s. Moreover, we assume DPAGs with bounded outdegree and indegree. Vertices are labeled by vectors of real numbers which either represent numerical or categorical variables. Given a vertex v in the DPAG, we give the following definitions:

- $\mathrm{out\_deg}(v)$ is the number of children of v;
- $\mathrm{in\_deg}(v)$ is the number of parents of v;
- $\mathrm{ch}[v]$ is the set of children of v, and $\mathrm{ch}_j[v]$ is the j-th child of v, with respect to P_v;
- $\mathrm{pa}[v]$ is the set of parents of v, and $\mathrm{pa}_j[v]$ is the j-th parent of v, with respect to P_v;
- $\boldsymbol{l}(v)$ is the input label associated to v, and $l_i(v)$ is the i-th element of the label;

Recursive neural networks [7] possess, in principle, the ability to memorize "past" information to perform structural mappings. The state transition function $\tau()$ and the output function $g()$, in this case, can be described by the following equations:

$$\begin{cases} \boldsymbol{x}(v) &= \tau(\boldsymbol{l}(v), \boldsymbol{x}(\mathrm{ch}[v])) \\ \boldsymbol{y}(v) &= g(\boldsymbol{l}(v), \boldsymbol{x}(v)) \end{cases} \tag{1}$$

where $\boldsymbol{x}(v)$ is the network state associated to vertex v, and $\boldsymbol{x}(\mathrm{ch}[v]) \equiv \boldsymbol{x}(\mathrm{ch}_1[v]), \ldots, \boldsymbol{x}(\mathrm{ch}_{\mathrm{out\_deg}(v)}[v])$. This formulation, however, is based on a structural version of the *causality* assumption, i.e., the output $\boldsymbol{y}(v)$ of the network at vertex v only depends on descendants of v. Specifically, RCC equations (1), where we disregard direct connections between hidden units, become

$$\begin{aligned} x_1(v) &= \tau_1(\boldsymbol{l}(v), x_1(\mathrm{ch}[v])) \\ x_2(v) &= \tau_2(\boldsymbol{l}(v), x_2(\mathrm{ch}[v]), x_1(\mathrm{ch}[v])) \\ &\vdots \\ x_m(v) &= \tau_m(\boldsymbol{l}(v), x_m(\mathrm{ch}[v]), x_{m-1}(\mathrm{ch}[v]), .., x_1(\mathrm{ch}[v])) \end{aligned} \tag{2}$$

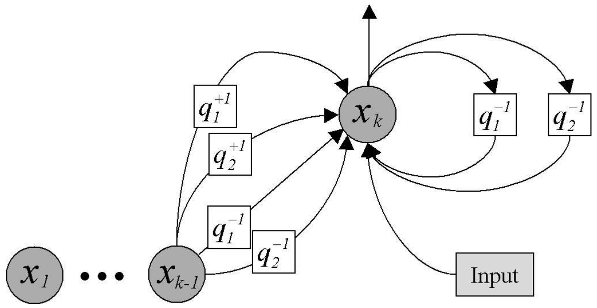

Fig. 1. Graphical model for x_k in CRCC, where $q_j^{-1}x(v) = x(\mathrm{ch}_j[v])$, and $q_j^{+1}x(v) = x(\mathrm{pa}_j[v])$.

where $x_i(v)$ is the i-th component of $\boldsymbol{x}(v)$, i.e., the output of the i-th hidden unit in the network. Since RCC is a constructive algorithm, training of a new hidden unit is based on already frozen units. Thus, when training hidden unit k, the state variables $x_1, \ldots, x_{k-1}$ for all the vertexes of all the DPAGs in the training set are already available, and can be used in the definition of x_k. Consequently, equations (2) can be expanded in a contextual fashion by using, where possible, the variables $x_i(\mathrm{pa}[v])$:

$$\begin{aligned} x_1(v) &= \tau_1(\boldsymbol{l}(v), x_1(\mathrm{ch}[v])) \\ x_2(v) &= \tau_2(\boldsymbol{l}(v), x_2(\mathrm{ch}[v]), x_1(\mathrm{ch}[v]), x_1(\mathrm{pa}[v])) \\ &\vdots \\ x_m(v) &= \tau_m(\boldsymbol{l}(v), x_m(\mathrm{ch}[v]), x_{m-1}(\mathrm{ch}[v]), x_{m-1}(\mathrm{pa}[v]), \\ &\qquad \cdots, x_1(\mathrm{ch}[v]), x_1(\mathrm{pa}[v])) \end{aligned} \tag{3}$$

which constitute the equations for the proposed Contextual Recursive Cascade Correlation (CRCC). A graphical model for x_k in CRCC is shown in Fig. 1. The Fig. 2 shows how adding new hidden units to the CRCC network leads to an increase of the "context window" associated to each vertex v. Specifically, the shown example focuses on the state computation of the vertex labeled "d" in the input tree, and describes for it, in a pictorial way, the functional dependences introduced by any new hidden unit inserted in the network. Unit 1 implements only causal computation. After adding unit 2, contextual information concerning the subtree rooted in the vertex labeled "g", contributes to the state definition of the vertex labeled "d". Finally, after adding unit 3, the context is extended to the whole tree.

Concerning the neural realization, the output x_k of the kth hidden unit over the current vertex v, in our contextual recursive cascade correlation network (CRCC) for structured data, can be computed as

$$x_k(v) \quad = \quad f(\mathrm{net}_k(v))$$

[1] If no supersource is present, a new vertex connected with all the vertexes of the graph with null *indegree* can be added.

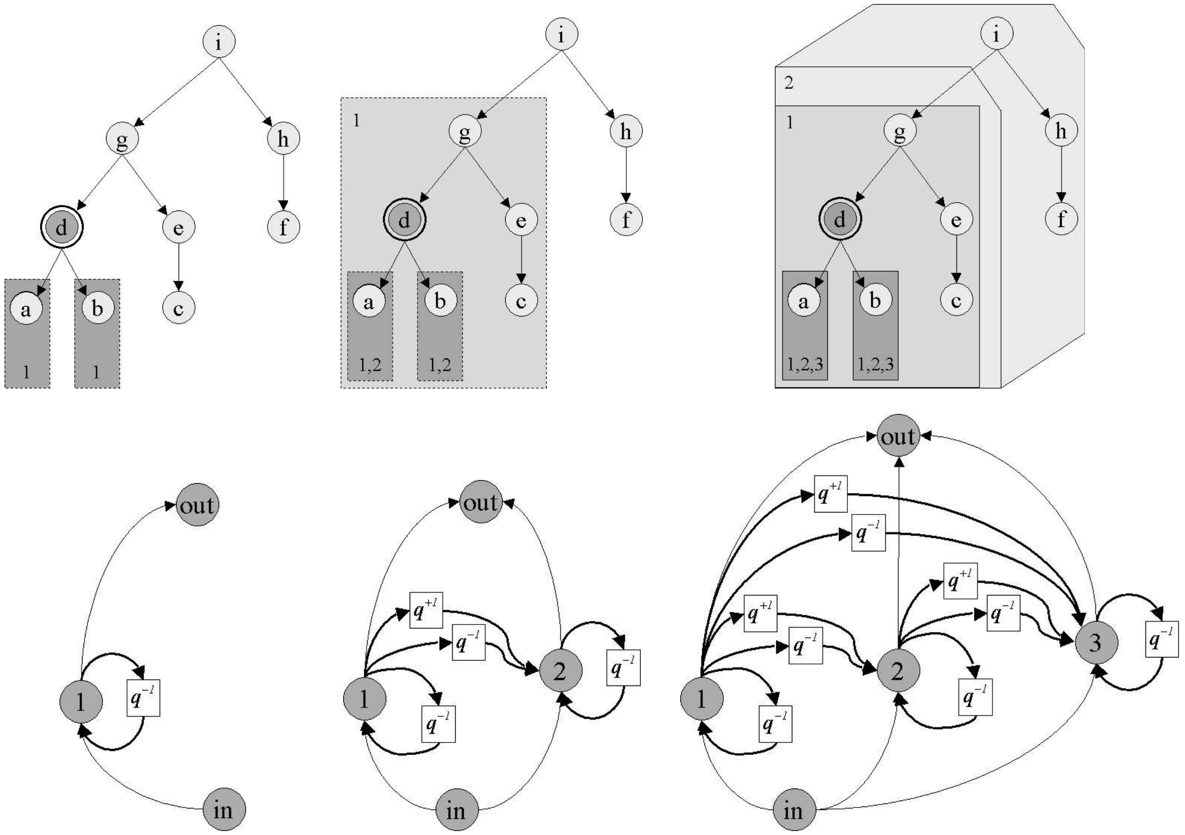

Fig. 2. Evolution of the "context window" for the vertex labeled "d" in the input tree with the growing of the network. The numbers associated to each box indicate the names of the units from which that information is coming, $\boldsymbol{q}^{-1} \equiv q_1^{-1}, \ldots, q_{\text{out\_deg}}^{-1}$ and $\boldsymbol{q}^{+1} \equiv q_1^{+1}, \ldots, q_{\text{in\_deg}}^{+1}$ (see Fig. 1).

$$\begin{aligned} \text{net}_k(v) \quad = \quad & \sum_{i=1}^{n} w_{ki} l_i(v) + & (4) \\ & \sum_{i=1}^{k} \sum_{j=1}^{\text{out\_deg}(v)} \hat{w}_{ki}^{j} x_i(\text{ch}_j[v]) + & (5) \\ & \sum_{i=1}^{k-1} \sum_{j=1}^{\text{in\_deg}(v)} \tilde{w}_{ki}^{j} x_i(\text{pa}_j[v]) & (6) \end{aligned}$$

where $f(\cdot)$ is a sigmoidal (or radial) function, w_{ki} is the weight of the i-th input element to the k-th hidden unit, $\hat{w}_{ki}^{j}$ is the weight of the edge connecting the i-th unit to the k-th current unit, which brings the encoded information of the j-th child of the current input vertex, and $\tilde{w}_{ki}^{j}$ is the weight of the edge connecting the i-th frozen unit to the k-th current unit, which brings the encoded information of the j-th parent of the current input vertex. Note that the first sum (4) corresponds to the "present" information, i.e., the label attached to v, the double sum (5) corresponds to the "past" information coming from descendants of v, while the double sum (6) corresponds to the "future" information coming from the subgraphs with supersource $\text{pa}_j[v]$. The network output function $g()$ (see Eq. 1) is implemented by one or more standard neurons.

Learning is performed as in standard Cascade Correlation by interleaving the minimization of the total error function (LMS) by a simple backpropagation training of the output layer, and the maximization of the (non-normalized) correlation, i.e. the covariance, of the new inserted hidden unit k with the residual error:

$$S = \sum_u \left| \sum_p (x_k(p) - \bar{x}_k)(E_u(p) - \bar{E}_u) \right| \qquad (7)$$

where u spans over the output units, p spans over all input patterns, and $\bar{x}_k$ is the mean output of the current unit, $E_u(p)$ is the residual error of the output unit u for the input pattern p, and $\bar{E}_u$ is the mean residual error of the output unit u.

The weight variation is then computed by the standard gradient ascent approach, deriving the equation (7) with respect to the desired weight:

$$\Delta w_{ki} = \eta \frac{\partial S}{\partial w_{ki}} = \sum_{u} \sigma_u \sum_{p} (E_u(p) - \bar{E}_u) \frac{\partial x_k(p)}{\partial w_{ki}} \quad (8)$$

where σ_u is the sign of the correlation between the output of the current hidden unit and the residual error of the output unit u.

Applying the RTRL algorithm approach as described in [9] we can determine the derivative of the output of the current hidden unit as follows:

$$\frac{\partial x_k(v)}{\partial w_{ki}} = \left(l_i(x) + \sum_{j=1}^{\text{out\_deg}(v)} \hat{w}_{kk}^{j} \frac{\partial x_k(\text{ch}_j[v])}{\partial w_{ki}} \right) f' \quad (9)$$

$$\frac{\partial x_k(v)}{\partial \hat{w}_{ki}^{h}} = \left(x_i(\text{ch}_h[v]) + \sum_{j=1}^{\text{out\_deg}(v)} \hat{w}_{kk}^{j} \frac{\partial x_k(\text{ch}_j[v])}{\partial \hat{w}_{ki}^{h}} \right) f' \quad (10)$$

$$\frac{\partial x_k(v)}{\partial \tilde{w}_{ki}^{h}} = \left(x_i(\text{pa}_h[v]) + \sum_{j=1}^{\text{out\_deg}(v)} \hat{w}_{kk}^{j} \frac{\partial x_k(\text{ch}_j[v])}{\partial \tilde{w}_{ki}^{h}} \right) f' \quad (11)$$

where f' is the first derivative of $f(\cdot)$. Note that equations (9), and (10) are the same used in standard RCC for structured data, while equation (11) is added so to include also contextual ("future" in sequences) information from frozen units. The above equations are recurrent and can be computed by observing that for all the leaves of the structured data (all vertexes with null outdegree) equation (9) becomes $\frac{\partial x_k(v)}{\partial w_{ki}} = l_i(v) f'$, and all remaining derivatives are null. Consequently, we only need to store the output values of the unit and its derivatives for each component of the structure.

III. EXPERIMENTAL RESULTS

In the following we report the results obtained with the CRCC (Contextual Recursive Cascade Correlation) model applied to the Quantitative Structure Property Relationship (QSPR) analysis of alkanes. In particular, we compare such results with the results obtained by the Recursive Cascade Correlation (RCC) on the same problem [2].

The problem consists in the prediction of the boiling point for a group of acyclic hydrocarbons (alkanes). For this problem, the causal model (RCC) has been proved to be competitive with respect to *ad-hoc* techniques (see [2]). In fact, the obtained results compares favorably versus the approach proposed by Cherqaoui et al. [3], which presents the *state-of-the-art* results. They apply a multilayer feed-forward neural network to a vectorial representation of alkanes able to retain the structural information which is known to be relevant to the prediction of the boiling point.

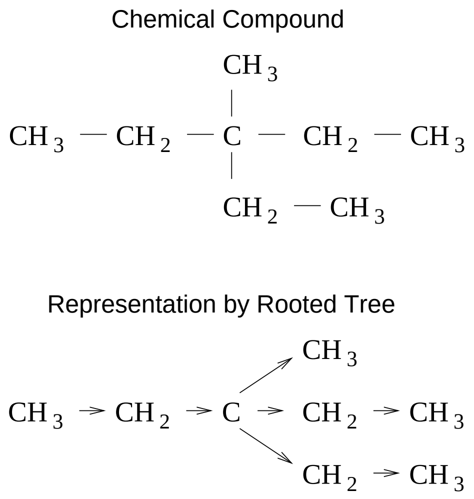
Chemical Compound
CH 3
CH 3 — CH 2 — C — CH 2 — CH 3
CH 2 — CH 3
Representation by Rooted Tree
CH 3
CH 3 → CH 2 → C → CH 2 → CH 3
CH 2 → CH 3

Fig. 3. Example of rooted-tree representation for an alkane (3-ethyl-3-methylpentane).

This task has been selected in order to have a direct comparison of the new approach (CRCC) with the standard causal model (RCC). Since the target property is related to global characteristic of the structures, such as the molecular size and the molecular shape, we believe that a model able to capture contextual information should improve the performance on this task.

On the other side the experiments allow to investigate the coherence of the causality assumption, and the effect of its relaxation, on a real-world application.

The data set used here, which is taken from [2], is based on all the 150 alkanes with up to 10 carbon atoms (C_nH_{2n+2}). Hydrogens suppressed graphs of alkane molecules are trees. Carbon-hydrogens groups are associated with vertexes, and bonds between carbon atoms are represented by edges. In order to represent them as rooted ordered trees, we used the I.U.P.A.C. nomenclature rules (a set of rules was developed in [2]). An example of alkane representation is shown in Fig. 3. The vertexes in the trees have a maximum outdegree of 3 and the maximum tree deth is 10. There is a total of 1331 vertexes in the data set.

The prediction of the boiling point yields to a regression task with a target associated to the root vertex of each tree. The target is the boiling point expressed in Celsius degrees (^{0}C) into the range $[-164, 174]$.

For the sake of comparison, we tested the prediction ability of the contextual versus the causal model using the same data set and learning parameters used for testing the causal model in [2]. Learning was stopped when the maximum absolute error for a single compound was

TABLE I
RCC vs CRCC.

	Causal RCC		**Contextual RCC**	
	Average (Min/Max)	Var.	Average (Min/Max)	Var.
Mean Abs. Train. Err.	2.09 (1.87/2.29)	0.01	1.84 (1.52/2.13)	0.05
Mean Abs. Test Err.	**3.15** (**2.50**/4.62)	0.41	**2.09** (**1.60**/2.38)	0.07
Max. Abs. Test Err.	9.91 (6.83/13.98)	6.62	5.71 (3.23/8.51)	2.61
Number of Units	164.38 (110/201)		159.88 (116/201)	

below 8 ^{0}C, or when a maximum number of hidden unit was reached (200 units for this set of experiments)[2]. All parameters have been chosen after an initial set of preliminary trials performed in order to determine an admissible range for the RCC models.

The data set we use here is composed of 135 compounds for training and 15 compounds for test. We repeated the training procedure 8 times in order to have a sound statistic. In table I the average, along with the best and the worst results, and the variance, over the 8 experiments, of the mean absolute error obtained for training and test set, the maximum absolute error on the test set, and the number of hidden units inserted in the model are reported. Specifically, the errors are expressed in ^{0}C.

In particular, it is possible to observe that the average of the mean test errors obtained by the contextual cascade correlation is around 1^{0}C less than the one obtained with the basic recursive cascade correlation. Moreover, the CRCC is also more stable than the RCC since the variance on the obtained test mean error is much lower then the RCC model. Finally, notice that the worst result obtained by the new model is still better than the best result of the basic model.

We observed that CRCC model gives better fitting and generalization results since, for each experiment, the mean test error is near the mean train error, and sometimes even better. This does not hold for the RCC. Moreover, even the average and the variance of the maximum error obtained with the CRCC experiments are much better than the RCC model (CRCC average and variance are around half of the RCC), and frequently, the CRCC maximum error in the test data set is lower that the maximum error in the training data set.

Moreover, notice that while improving the efficacy on the error results with the new model, the efficiency does not decrease, since the number of inserted hidden units by the two models, at the end of the training phase, is comparable.

In Fig. 4 and 5 the learning curves for two experiments are reported. In particular, the plot in Fig. 4 shows the behavior of the two cascade models in the best trial, reporting the mean training and test absolute errors obtained by the two models during learning as a function of the number of inserted hidden units. Instead, the plot in Fig. 5 reports the same information for the two cascade models ranked 4th (median) among all the results. These curves show that also the dynamical behavior of the CRCC is better than the dynamical behavior of the standard RCC. Moreover, it can be noted that the CRCC model can obtain the same fitting or generalization results of the RCC model, with much less hidden units. Specifically, the generalization results obtained by the RCC at the end of the training phase, can be obtained by CRCC using less then half units.

IV. CONCLUSION

We presented an extension of the RCC model based on the contextual analysis of structured domains. Using the CRCC model we afforded a real-world task, i.e. the prediction of the boiling point of a set of alkanes, in order to show the improvements that can be obtained using a contextual approach versus a pure causal approach (standard RCC). It should be noted that, even if the prediction task was defined only on root vertexes, the CRCC model was able to develop internal representations taking into account the context in which each subtree occurs. This cannot be done by RCC, for which only a single internal representation for each subtree can be developed. Moreover, the possibility to have information about the context in CRCC opens new ways to the error gradient flow through the structures, improving the efficacy of the gradient descent process.

The successful results suggest that the new model can be adopted as an alternative to the basic RCC whenever it is not possible to guarantee the soundness of the causal assumptions for the domain under analysis.

References

[1] P. Baldi, S. Brunak, P. Frasconi, G. Pollastri, and G. Soda. Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics*, 15(11):937–946, 1999.

[2] A.M. Bianucci, A. Micheli, A. Sperduti, and A. Starita. Application of cascade correlation networks for structures to chemistry. *Journal of Applied Intelligence (Kluwer Academic Publishers)*, 12:117–146, 2000.

[3] D. Cherqaoui and D. Villemin. Use of neural network to de-

[2] Actually, for few trials the maximum number of hidden units is reached before the maximum error on the training data set was below 8 ^{0}C. However, we found that in such cases, both the mean error and the maximum error on the training data set are comparable to the values obtained with the trials that respect the stopping criterion on maximum train error.

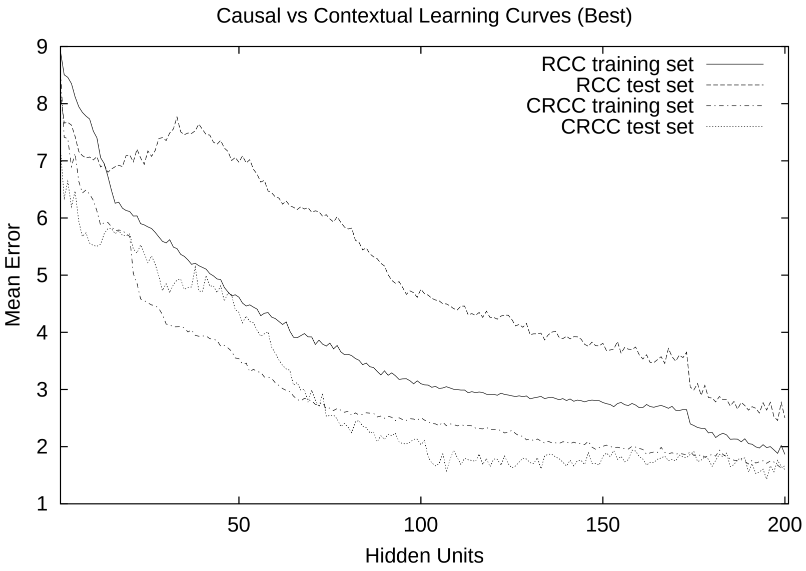

Fig. 4. Comparison of the best trial's learning curves for RCC and CRCC models. The mean training and test errors are plotted versus the number of inserted hidden units.

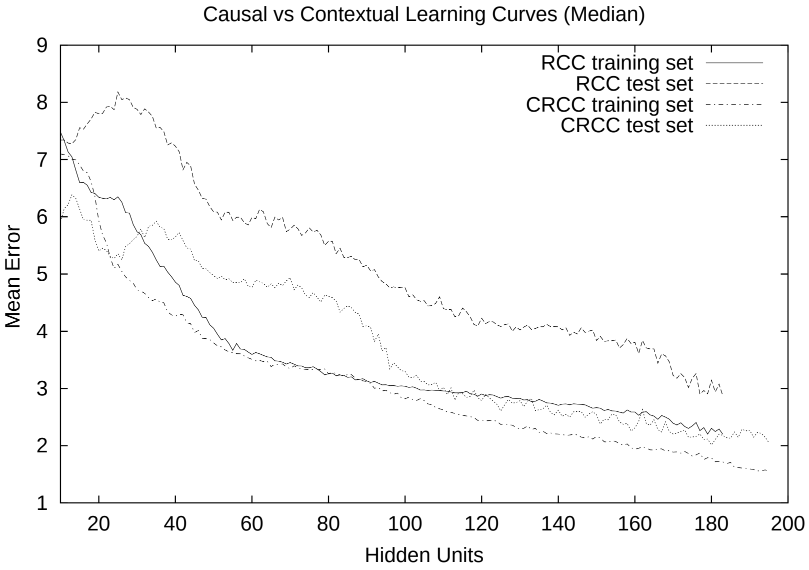

Fig. 5. Comparison of the learning curves obtained for the 4th best trial for RCC and CRCC models.

termine the boiling point of alkanes. *J. Chem. Soc. Faraday Trans.*, 90(1):97–102, 1994.

[4] 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.

[5] A. Micheli, D. Sona, and A. Sperduti. Bi-causal recurrent cascade correlation. In *Proc. of the Int. Joint Conf. on Neural Networks - IJCNN'2000*, volume 3, pages 3–8, 2000.

[6] 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.

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

[8] H. Wakuya and J. Zurada. Bi-directional computing architectures for time series prediction. *Neural Network*, 14:1307–1321, 2001.

[9] R. J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1:270–280, 1989.