ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 269-274

Segmental Duration Control by Time Delay Neural Networks with Asymmetric Causal and Retro-Causal Information Flows

Çağlayan Erdem, Hans Georg Zimmermann
Siemens AG , Corporate Technology, D- 81730 Munich, Germany

email: {Caglayan.Erdem,Georg.Zimmermann}@mchp.siemens.de

Abstract. The generation of pleasant prosody parameters is very important for speech synthesis. A Prosody generation unit can be seen as a dynamical system. In this paper sophisticated time-delay recurrent neural network (NN) topologies are presented which can be used for the modeling of dynamical systems. Within the prosody prediction task left and right context information is known to influence the prediction of prosody control parameters. This can be modeled by causal-retro-causal information flows [1]. Since information being available during training is partially unavailable during application, there is a structural switching from training to application. This structural change of the information flow is handled by two asymmetric architectures. These proposed new architectures allow the integration of further a priori knowledge. By this we are able to improve the performance of our duration control unit within our text-to-speech (TTS) system *Papageno*.

1 Introduction

Our acoustic prosody module consists of a duration control and a f0-contour unit. Both are modeled by NN (see [2]). There are also rule based duration control methods [3], which depending on rules modifies the duration of a segment by a multiplicative or additive scaling factor. Appropriate segmental durations are very important for a natural sounding synthetic voice. A duration control module with low performance has a very strong impact on the f0-contour unit. Similar to the f0-contour prediction task [1] the duration control unit uses left (past) and right (future) contextual information to establish the prediction. The left contextual information is the text being already read while the right contextual information is given by the text to read next. A segmental duration module has to control the rhythm of a synthetic voice and the known effect of final lengthening. So local and global structures have to be mapped. The state-of-the-art causal-retrocausal modeling was presented in [1] for the f0-prediction task. The shortcomings of that architecture are a fix-point recurrences causing stability problems during training, and a non-observance of the mentioned structural switching. The causal-retrocausal-error-correction (CRCEC) NN architecture is used as a basis for the modeling of the duration control task. Different architectures will be presented to overcome these problems.

2 CRCECNN

The CRCECNN architecture is depicted in Fig. 1 for one time step (solid lines). This architecture uses shared weights and has a symmetrical extension to the neighboring time steps (dotted lines). As can be seen there are two different information flows. In the upper part of Fig. 1 there is a causal information flow denoted by the matrix A carrying state information (s_{t+i}) of the dynamics between neighboring state clusters. This path allows the mapping of long-term forecasts. A retro-causal information flow (r_{t+i}) is given by the matrix F in the lower part of this Figure. Within each time step i there are two error-correction (see [4]) parts incorporated. Both are coupled by the usage of one output cluster (z_{t+i}). The error-correction will be explained using the causal information flow path. While matrix B introduces external information u_t to the system, the matrix C transforms the state s_t to its expectation y_t. D propagates the model error (the expectation y_t being compensated by the observation y_t^d) to cluster s_{t+1}. The path

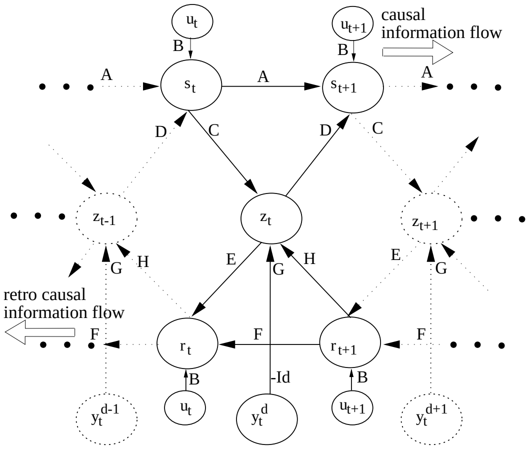

Figure 1: Causal-Retro-Causal Modeling.

$$s_{t-1} \rightarrow C \rightarrow z_{t-1} \rightarrow D \rightarrow s_t$$

allows to map local structures as shocks or short term effects. The z-clusters represent the output clusters of the NN architecture. In cluster z_t the difference z_t= Cs_t - y_t^d (forecast error) between the expectation of the NN and the observation y_t^d is computed. Note that y_t^d is propagated by $-Id$ to z_t. This difference has its optimum in zero, since this denotes no forecast error. Having no forecast errors results in a perfect description of the dynamics. So the target vectors (z_{t+i}) are set to zero during training. If there is no mismatch between expectation and observation then no further information is propagated by matrix D to state s_t and we almost obtain a simple finite unfolding NN. An existing mismatch delivers further input information to state s_{t+1}. This information is used during training for the adaption of parameters. By this error correction principle we obtain in z_t an internal vector driving the transition of the system state together with external input u_t and previous states. These internal vectors generate the error flow. It is computed at each output cluster time step of the unfolding. If the internal autoregressive part coded in A and all external driving forces of a dynamics are known, it would be possible to give a perfect description of the dynamical system. But if it is not possible to identify the dynamics due to missing or unknown externals or noise, the last model error is an indicator of the models misspecification. Since the model error is used as a measure of unexpected shocks, the learning of false dependencies is lowered and models generalization ability is improved [5]. Incorporating information flow from the right to the left captures retro-causal dependencies. If this is handled symmetrically over all time steps this modeling results in fix point recurrencies as depicted in [1]. One closed loop is given by:

$$s_t \rightarrow C \rightarrow z_t \rightarrow E \rightarrow r_t \rightarrow H \rightarrow z_{t-1} \rightarrow D \rightarrow s_t$$

CRCECNN optimally fits to the information flow of the application, but makes training hard to solve. So substructures are to find, which overcome the closed-loop problem. Therefore this paper proposes a partial symmetric expansion in the following subsection which results in a partial CRCECNN (P-CRCECNN).

3 P-CRCECNN

The NN depicted in Figure 2 utilizes shared weights and finite unfolding. The coupling of both information flows is realized by only one output cluster z_t instead of the coupling at each time step within CRCECNN. By coupling those information flows within the present time step this new architecture does not contain fix-point recurrent loops, which might cause instabilities during training. In the following this architecture will be used for further adaptations.

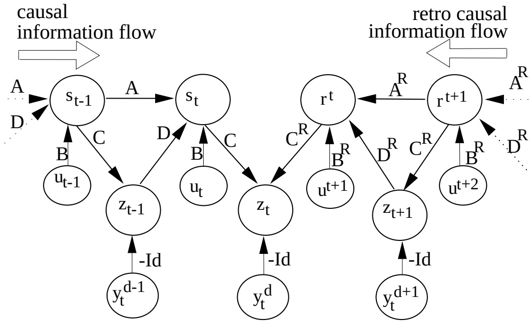

Figure 2: P-CRCECNN

4 Structural Switching

The structural switching will be explained using Fig. 2. During training all segmental durations modeled as observations y^d_{t+i} are known. But within the application there are no observation available for $i \geq 0$, because they are not predicted yet. For $i < 0$ predictions of the NN are re-utilized as observations. Because of this mismatch between training and application the retrocausal information flow has to be treated in a specific way. In the following two different ways of asymmetric P-CRCECNN are explained which overcome this mismatch. In Fig. 3 the idea of removing connections after training is depicted. The dotted connections C^R and D^R are trained. So the architecture is the same as depicted in Fig. 2 during training. But within the application connections C^R and D^R are removed. The resulting architecture is then a finite unfolding in time without the error correction principle for the retro-causal information flow during application. The next architecture is established by using finite unfolding in time for the retro-causal path during training and application as shown in Fig. 4.

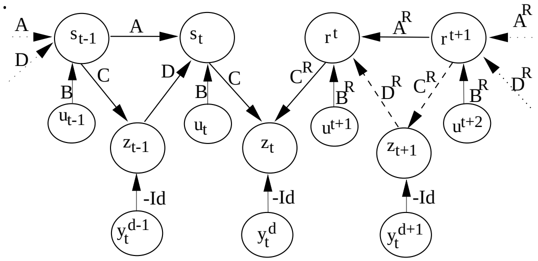

Figure 3: P-CRCECNN_removed

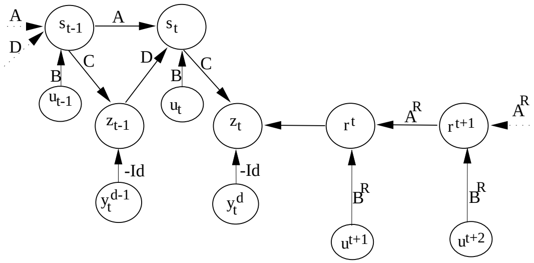

Figure 4: P-CRCECNN_finUnfold

5 Application and Results

In this section the application of asymmetric P-CRCRECNNs within the segmental duration control unit of our acoustic prosody module is presented. These data-driven methods are applied to recordings of three hours of a german news speaker reading news from *Frankfurter Allgemeine Zeitung*. The patterns for training (80%) and testing (20%) are separated. A validation set of (20%) is selected randomly from the training set. This database is the same as used within the f0-generation task [1]. The f0-generation task utilized patterns organized on syllable level. But within this task patterns are organized

on triphone level. Following information is presented to the NN in a context of seven phonemes to the left and right.
a) phonetic information: with one-out-of-n coding the phoneme index is presented here. A phoneme-set of 45 phonemes is used. Additionally the four phoneme classes (vowel, fricative, nasal, liquid, and plosive) are presented here.
b) positional information: Discrete information denotes whether the according syllable is an initial, medial or final one within the phrase and the word. Continuous information is given by the relative syllable position within a sentence and phrase.
c) stress information: Flags denoting the stress type of the according syllable are coded here. Word level stress is presented by four flags. Sentence level stress consists of two stress marks.
d) linguistic categories: An one-out-of-n (set of 14 categories) coded linguistic category denotes the category type of the according word.
These inputs are presented at each time step of the unfolding clusters denoted by u_{t+i}. The according output vectors are modeled as observations and are presented at each time step in the clusters denoted by y^d_{t+i}.

NN	MSE in %	correlation	marks
P-CRCECNN_removed	3.49	0.8723	3.5
P-CRCECNN_finUnfold	4.35	0.8232	2.5

Table 1: results

Target values for the NN are normalized to ensure an optimized signal-flow during training of the NN due to *tanh*-activation-function within the causal and retro-causal state clusters. A first normalizing of segmental duration is obtained by the mean and standard deviation value from the used triphon classes. A second normalizing was necessary to ensure an optimized signal flow during training of the NN. The mean and standard deviation were derived from the first normalized segmental durations. CRCECNN are hard to train, therefore we changed our intention to train only cut down architectures. For evaluation the trained NN were used to predict segmental durations of sentences which were in the test set. Three audio-files are generated with those predictions which are then used for evaluation. In table 1 the mean square errors (MSE) are given for the two asymmetric NN realizations and it depicts the correlation of the predictions and the original segmental durations. Within both experiments the P-CRCECNN_removed method gives better results. In a further test it was observed that 85.6% of phrase-breaks were realized with a clear final lengthening. As perception is a highly complex process not necessarily modeled appropriately by isolated physically distances, informal listening test were performed. Files generated by resynthesis utilizing the three different methods were presented to seven non-expert listeners. They had to judge which of the presented files were most pleasant and least pleasant to them. They also had to give a ranking. This ranking was then scaled on a value set from 4 to

[0] A triphone is a phoneme considering its predecessor and successor phonee. So same phonemes with different predecessor or successor result in different triphones

1, with 4 denoting the most pleasant file. The mean values of the ranking are shown in table 1. The asymmetric P-CRCECNN with connections removed after training was evaluated to be most pleasant. This architecture uses the error correction principle within the retro-causal path for modeling local prosodic structures. This seems to help the long term forecast path improving its generalization ability, as this NN performs better than the P-CRCECNN_finUnfold, which does not utilize error correction. The long term forecast path within P-CRCECNN_finUnfold has also to capture short time events.

6 Conclusions

In this paper specialized error correction NN architectures are presented. First a partial CRCECNN is proposed to overcome closed loop computation problems. Afterwards a structural change of the information flow from training to application is described and integrated into modeling of the NN. Two asymmetric NN architectures are presented to overcome this structural switching. They are applied within the duration control task of our TTS system *Papageno.* The best performing NN is selected by numeric and listening experiments. Informal listening tests show that an asymmetric architecture improves performance. A sophisticated architecture with connections trained but removed afterwards during application delivers most pleasant prosodic parameters, as this architecture is the closest one to the information flow of the duration control task.

References

[1] H.-G. Zimmermann, Achim F. Müller, Çağlayan Erdem, and Rüdiger Hoffmann : prosody generation by causal retro-causal error correction neural networks , *MSC 2000, Japan, ATR*

[2] W.N. Campbell: Syllable-based segmental duration

in G. Bailly and C. Benoit, editors., Talking Machines: Theories, Models and Designs, pages 211-224, Elsevier, North-Holland, 1992

[3] D.H Klatt: Review of text-to-speech conversion for English *Journal of the Acoustical Society of America, 82(3), 737-793*

[4] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstucture of Cognition*, volume I, chapter Foundations, pages 318–362. MIT Press/Bradford Books, Cambridge, MA, 1986.

[5] H. G. Zimmermann, R. Neuneier, R. Grothmann: Modeling of Dynamical Systems by Error Correction Neural Networks, *in: A. Soofi and L. Cao [eds.], Modeling and Forecasting Financial Data, Techniques of Nonlinear Dynamics, Kluwer Academic Publishers, 2000*