

Duration Control by Asymmetric Causal Retro-Causal Neural Networks

Çağlayan Erdem^(1,2) & Hans Georg Zimmermann⁽²⁾

(1) Dresden University of Technology, D-01062 Dresden, Germany

(2) Siemens AG , Corporate Technology, D- 81730 Munich, Germany

{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

Different Methods for duration control have been proposed. In [2] a rule based duration control method was presented, which depending on rules modifies the duration of a segment by a multiplicative or additive scaling factor. [3] differentiates between different duration models. A data-driven method with good generalization abilities is presented in [4] using a NN for the syllable duration control.

Our acoustic prosody module consists of a duration control and a f0-contour unit. In a first approach our duration control module used a *classification and regression trees (CART)* (CART) like method. The parameters for the nodes are derived from triphon clusters obtained by a tree-based clustering algorithm provided by standard clustering within HTK [5]. This approach will not be explained in detail. In the following it is called *stat*.

In a first run this statistic method was enhanced by a more sophisticated statistical approach considering larger contextual information on syllable and word level. But there is a dilemma between robustness and significance of the statistics as *Papageno* utilizes a restricted speech database for its fundus (about 1k sentences). Therefore the NN approach is employed to solve this dilemma due to his generalization property.

As depicted in Fig. 1 now both units are modeled by NN. In *Papageno* the segmental duration control is handled first. Those segmental durations generated by the duration control unit are afterwards used as inputs to the f0-generation unit.

Continuous positional information of syllables is derived from these durations, which are important for the f0-generation task. A segmental duration module has to control the rhythm of a synthetic voice and the known effect of final lengthening.

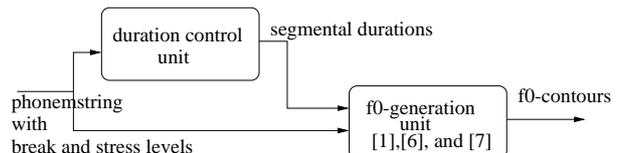


Figure 1: acoustic prosody unit

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 input features of both modules are very similar. They are organized on syllable level for the f0-contour prediction and on phoneme (triphone) level for the duration control task. The state-of-the-art causal retro-causal modeling was presented in [1] for the f0-prediction task. The shortcomings of that architecture are a fix-point recurrence, that causes stability problems during training, and a non-observance of the mentioned structural switching. The causal retro-causal-error-correction (CRCEC) NN architecture [1] is used as a basis for the modeling of the duration control task. Different architectures will be presented to overcome these two problems.

This paper consists of four parts. In section 2 basics (finite unfolding, error correction) and a new partial retro-causal expansion for the integration of the structural switching are discussed. In section 4 asymmetric architectures are discussed which overcome the structural switching of the information flow. Section 5 depicts the application of the asymmetric modeling and gives results.

2. CRCECNN

The state-of-the-art causal retro-causal error-correction (CRCEC)-NN modeling was proposed in [1]. The according architecture is depicted in Fig. 2 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. 2 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 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

$$s_t \rightarrow C \rightarrow z_t \rightarrow D \rightarrow s_{t+1}$$

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 $z_t = C s_t - y_t^d$ (forecast error) between the expectation of the NN and the observation y_t^d is computed. Note that y_t^d is propagated by $-Id$ (identity) 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 vector (z_t) is set to zero during training.

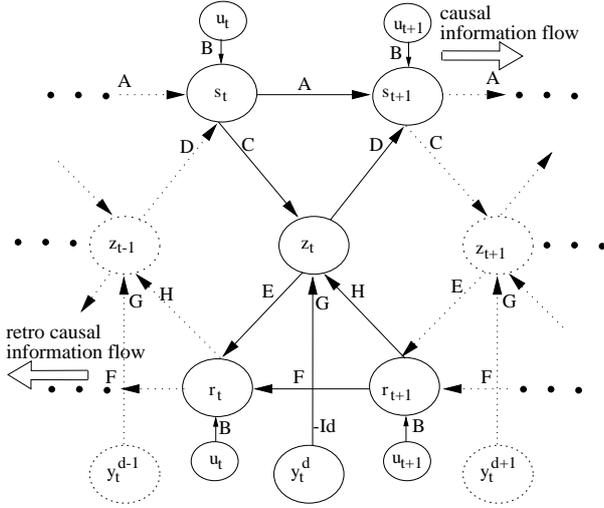


Figure 2: Causal Retro-Causal Modeling.

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 adaptation 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 [8] [9].

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 recurrences as depicted in [1]. One closed loop is given by :

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

These fix point recurrences complicate computation of the resulting CRCECNN. 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 3 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.

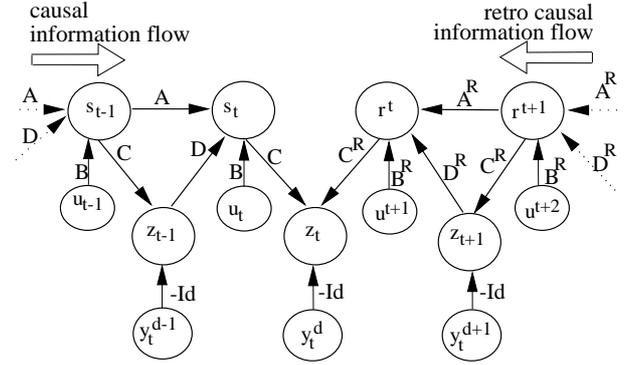


Figure 3: P-CRCECNN

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.

4. Structural Switching

The structural switching will be explained using Fig. 3. During training all segmental durations modeled as observations y_{k+i}^d 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 retro-causal information flow has to be treated in a specific way. In the following to different ways of asymmetric P-CRCECNN are explained which overcome this mismatch.

In Fig. 4 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. 3 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. 5.

5. Application and Results

In this section the application of asymmetric P-CRCECNNs 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

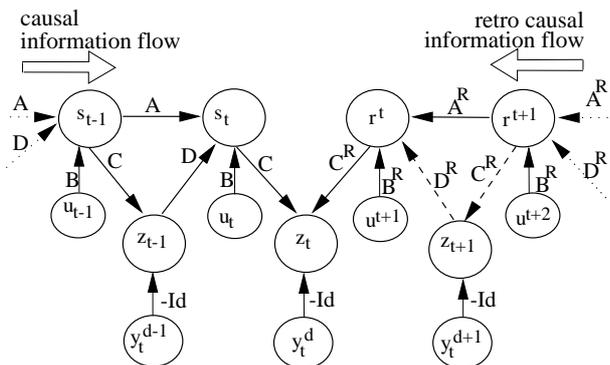


Figure 4: P-CRCECNN_removed.

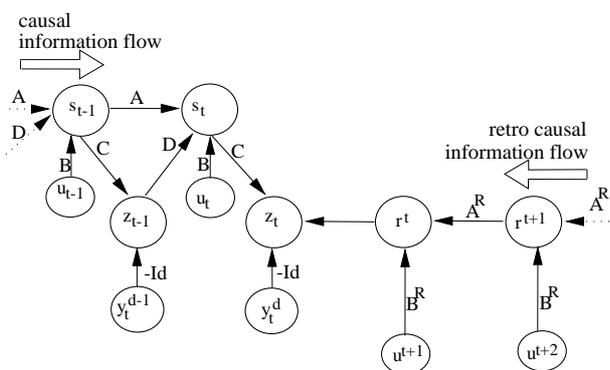


Figure 5: P-CRCECNN_finUnfold

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.

- 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.
- 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.
- 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.
- 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_{t+i}^d . 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 triphone 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.

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 deviations and root-mean-square-error (rmse) (computing generated durations and original duration in the database) are given in milliseconds for the used methods.

NN	standard deviation	rmse
stat	13 ms	18.1 ms
P-CRCECNN_removed	11.9 ms	16.4 ms
P-CRCECNN_finUnfold	14.9 ms	19 ms

Table 1: deviations

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 different presented NN-Architectures and the original one 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 1, with 4 denoting the most pleasant file.

The mean values of the ranking are shown in table 2. 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

method	marks
stat	1.3
P-CRCECNN_removed	3.5
P-CRCECNN_finUnfold	2.5

Table 2: listening tests

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 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.

7. References

- [1] Zimmermann, H.-G.; Müller, Achim F.; Erdem, Çağlayan; Hoffmann, Rüdiger, 2000. Prosody Generation by Causal Retro-Causal Error Correction Neural Networks. *MSC, Japan, ATR*, 116-121.
- [2] Klatt, D., 1987. Review of text-to-speech conversion for English. *Journal of the Acoustical Society of America*, 82 (3), 737-793.
- [3] van Santen, J.P.H; Olive, J.,1990. The Analysis of Contextual Effects on Segmental Duration. *Computer Speech & Language*, 4, 359-390.
- [4] Campbell, W.N., 1992. Syllable-based segmental duration. in G. Bailly and C. Benoit, editors., *Talking Machines: Theories, Models and Designs*, Elsevier, North-Holland, 211-224.
- [5] Holzapfel, Martin, 1999. HMM based database segmentation and unit selection for concatenative Speech Synthesis. *Proc.joint ASA/DAGA spring meeting hosted by DAGA*, Berlin.
- [6] Erdem, Çağlayan; Holzapfel Martin; Hoffmann, Rüdiger, 2000. Natural f0-contours with a new Neural-Network-hybrid-approach. In *ICSLP*, Beijing, Vol 3, 227-230.
- [7] Haury, R.; Holzapfel, M., 1998. Optimisation of a Neural Network for Pitch Contour Generation. *ICASSP*, Seattle, Vol 1, 297-300.
- [8] Zimmermann, H. G.; Neuneier, R.; Grothmann, R., 2002. Modeling of Dynamical Systems by Error Correction Neural Networks. In: A. Soofi and L. Cao [eds.], *Modeling and Forecasting Financial Data, Techniques of Non-linear Dynamics*, Kluwer Academic Publishers, forthcoming.
- [9] Rumelhart, D. E.; Hinton, G. E.; Williams, R. J., 1986. Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Volume 1, chapter Foundations, MIT Press/Bradford Books, Cambridge, MA, 318-362.