

Weighted Neural Network Ensemble Models for Speech Prosody Control

Harald Romsdorfer

Speech Processing Group, ETH Zurich, Switzerland

romsdorf@tik.ee.ethz.ch

Abstract

In text-to-speech synthesis systems, the quality of the predicted prosody contours influences quality and naturalness of synthetic speech. This paper presents a new statistical model for prosody control that combines an ensemble learning technique using neural networks as base learners with feature relevance determination. This weighted neural network ensemble model was applied for both, phone duration modeling and fundamental frequency modeling. A comparison with state-of-the-art prosody models based on classification and regression trees (CART), multivariate adaptive regression splines (MARS), or artificial neural networks (ANN), shows a 12% improvement compared to the best duration model and a 24% improvement compared to the best F_0 model. The neural network ensemble model also outperforms another, recently presented ensemble model based on gradient tree boosting.

Index Terms: speech synthesis, prosody control, neural networks, ensemble models

1. Introduction

The prosody of a speech signal can be described at the perceptual level in terms of *pitch*, (*sentence melody*), (*speech rhythm*), and *loudness*. The physically measurable quantities by which speech segments can be modified are the acoustic parameters *fundamental frequency* (F_0), *segment duration*, and *signal intensity*. The prosody control component must generate these physical parameters using linguistic features obtained from the input text.

Given prosody corpora of limited size, the statistical models that are applied in current TTS synthesis systems, like CART-, MARS-, HMM-, or ANN-based generation models, have one or more of the following disadvantages: their *prediction accuracy* is too low to achieve natural sounding prosody. Their *generalization capability* for input factor combinations that are not covered by the prosody corpora is not good enough. Or they are not *robust* enough against outliers of acoustic parameters due to errors in the segmentation, in the labeling, or in F_0 extraction of the prosody corpora.

A comparison of these models in [1] showed that CART-based models are too inaccurate. MARS-based models achieve good accuracy, but have, according to [2], a lower generalization capability and are sensitive to outliers. ANN-based models show good generalization capabilities and also good accuracy, but need in general more training data than the other methods to reach equal accuracy and are, in case of limited training data, also sensitive to outliers that can result in unsteady output contours.

Recent advances in machine learning show that weighted ensembles of ANNs for regression can significantly improve prediction accuracy, generalization capability, and robustness against outliers when compared to single networks, cf. [3].

Factor relevance determination procedures can be used to remove (or “prune”) less relevant input factors. Thus, the problem of small prosody corpora is less critical, and certain statistical models, like ANNs, also gain interpretability, as irrelevant input factors are removed.

2. Weighted Neural Network Ensembles

A weighted ensemble is a set of independently trained statistical models, that are then often called *base learners*. The prediction output of the ensemble is a linear combination of the prediction outputs of the individual models

$$Y_M(\mathbf{x}) = \sum_{m=1}^M w_m \mathbf{y}_m(\mathbf{x}) \quad (1)$$

where M is the number of individual models, \mathbf{y}_m is the prediction output of the m -th member, and w_m is a decreasing function of the prediction error of the m -th member over the whole training set. Thus, each ensemble member is weighted according to its individual performance.

2.1. Base Learners

As *base learners*, the author applied feed-forward ANNs for segment duration prediction, and recurrent ANNs (RNNs) for predicting F_0 contours. A feed-forward neural network, also known as multilayer perceptron (MLP), can be described as a series of functional transformations that are represented in case of two layers of weights by the network function

$$y_k(\mathbf{x}, \mathbf{w}) = g \left(\sum_{j=0}^M w_{kj}^{(2)} h \left(\sum_{i=0}^D w_{ji}^{(1)} x_i \right) \right) \quad (2)$$

and in case of three layers of weights by the network function

$$y_k(\mathbf{x}, \mathbf{w}) = g \left(\sum_{r=0}^N w_{kr}^{(3)} h \left(\sum_{j=0}^M w_{rj}^{(2)} h \left(\sum_{i=0}^D w_{ji}^{(1)} x_i \right) \right) \right) \quad (3)$$

where the set of all weight and bias parameters have been grouped together into a vector \mathbf{w} . $w_{j0}^{(1)}$, $w_{k0}^{(2)}$, $w_{r0}^{(2)}$, and $w_{k0}^{(3)}$ represent the bias parameters of the individual weight layers. $h(\cdot)$ is a differentiable, nonlinear activation function of the hidden units. $g(\cdot)$ is a differentiable activation function of the output units. Thus, the neural network is simply a nonlinear function from a set of input variables $\{x_i\}$ to a set of output variables $\{y_k\}$ controlled by a vector \mathbf{w} of adjustable parameters.

In all experiments, $h = \tanh$ was used as hidden unit activation function for all networks. The output unit activation function $g(\cdot)$ was the identity function.

Input and output values are rescaled by applying a linear normalization using mean and variance calculated with respect to the training set. Thus, the transformed variables have zero

mean and unit standard deviation. The linear rescaling makes it possible to initialize network weights by random selection from a zero mean, unit variance isotropic Gaussian where the variance is scaled by the fan-in of the units as appropriate. Network weights of feed-forward ANNs are trained using the well-known *error back-propagation* procedure of [4]. RNNs are trained using *error back-propagation for sequences*, as described, e.g., in [5]. Error back-propagation for sequences treats RNNs as feed-forward ANNs by “unfolding” them in time. In both procedures, the *scaled conjugate gradient* algorithm introduced in [6] is applied for optimization using a sum-of-squares error function of network outputs.

2.2. Weighting Functions

As *weighting functions*, the author tested an exponential (4) and a potential weighting function (5), that were suggested in [3], and for comparison also the arithmetic mean (6)

$$w_i = \frac{\exp(-\alpha e_i)}{\sum_{j=1}^M \exp(-\alpha e_j)}, \quad (4)$$

$$w_i = \frac{e_i^{-\alpha}}{\sum_{j=1}^M e_j^{-\alpha}}, \quad (5)$$

$$w_i = \frac{1}{M}, \quad (6)$$

where e_i is the prediction error of the i -th model. M is the number of individual models in the ensemble. α is a weighting coefficient. Setting $\alpha = 0$ results in the arithmetic mean for both weighting functions.

The prediction error is calculated using the *normalized mean squared error (NMSE)*. The NMSE is defined as the mean squared error of the predictor \mathbf{y} on the test set T divided by the variance σ_D^2 of the complete data set D , cp. [3], as

$$NMSE_T(\mathbf{y}) = \frac{\frac{1}{N} \sum_{i=1}^N (t_i - \mathbf{y}(\mathbf{x}_i))^2}{\sigma_D^2}, \quad (7)$$

with a test set T consisting of N input/target value pairs $\{\mathbf{x}_i, t_i\}$.

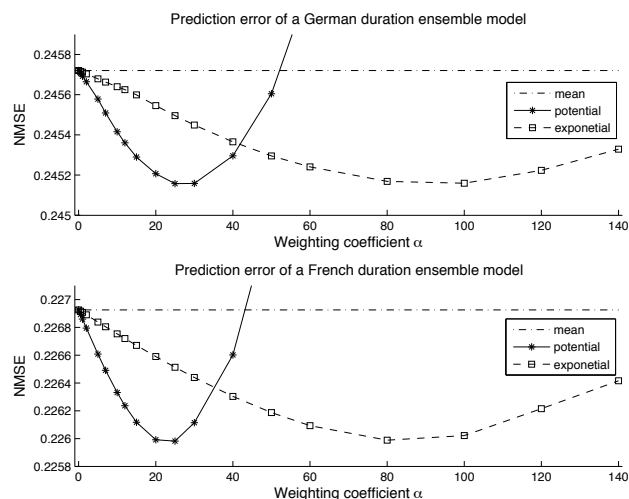


Figure 1: Normalized mean squared error as a function of the weighting coefficient α using arithmetic mean, potential weighting, and exponential weighting function for German (above) and for French (below) female duration prediction.

As an example for the influence of the weighting function on prediction error, Figure 1 shows the normalized mean squared error as a function of the weighting coefficient for German and for French segment duration prediction. For large values of α , overfitting is observed, since only a few particular networks contribute to the ensemble output. Exponential weighting results in more robust error curves than potential weighting. Therefore, exponential weighting with a weighting coefficient $\alpha = 80$ was used for German and for French ensemble construction in all experiments.

2.3. Network Aggregation

[3] present an evaluation of aggregation methods for ANN ensembles on several standard regression problems. The tested aggregation methods base either on *Boosting*, that was introduced for classification problems in [7], and later extended for regression problems, e.g., in [8] and in [2], or they base on *Bagging* (short for “bootstrap aggregation”), which was introduced in [9]. A *Weighted Bagging* algorithm (W-Bagging) and the so-called *Weighted Stepwise Ensemble Construction Algorithm* (W-SECA), another bagging-like algorithm, both introduced in [10], were among the top performers in nearly all test cases. These weighted bagging-based algorithms outperformed the boosting methods and also other regression methods that based, e.g., on Support Vector Machines, which were introduced in [11], or on Boosting using Radial Basis Functions networks, which are described, e.g., in [12]. For aggregation to be effective, however, the individual networks of the ensemble must be *both accurate and diverse*.

Diverse individual networks can be obtained, e.g., by varying the internal network structures or by using different adequately-chosen subsets of the training set to optimize the parameters of the individual networks. A vital element of success when using different subsets is the instability of the learning algorithm, cf. [9]. ANNs are therefore well suited, as this instability comes naturally from the inherent randomness of the training algorithms of ANNs.

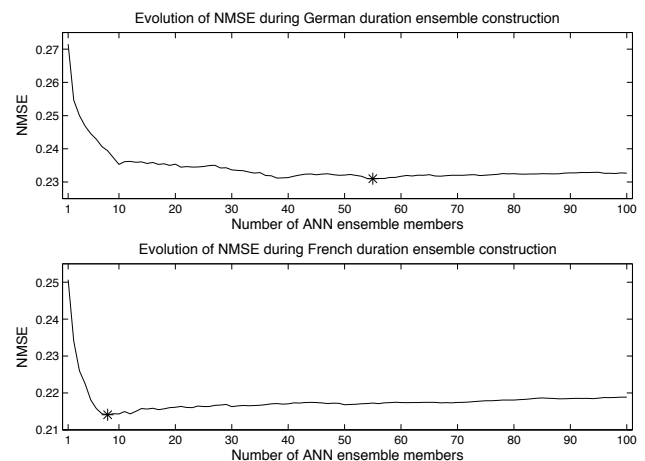


Figure 2: Normalized mean squared error as a function of the number of ensemble members during duration ensemble construction. The ensemble model with lowest prediction error is indicated by a star. The best ensemble for German female duration prediction consists of 55 ANNs. The best French female duration ensemble model has 8 ANN members.

German duration ensemble								
Network Nr.	1	2	3	4	5	6	7	8
Factors	55	60	60	61	55	61	67	66
Layer 1	30	25	30	25	30	23	50	50
Layer 2						5		

French duration ensemble								
Network Nr.	1	2	3	4	5	6	7	8
Factors	80	70	60	60	58	56	68	60
Layer 1	20	15	16	20	16	20	14	17
Layer 2		15	20	10	20	10	20	15

Table 1: Network structure of each ANN member of the best ensemble for German female and for French female duration control shown in Fig 2. For each ensemble member, the number of input factors and the number of nodes of the first and of an optional second hidden layer is given.

Network accuracy can be increased, e.g., by fitting the complexity of the network and the size of the training set optimally to each other. However, increasing the size of a prosody corpus is very elaborate and often not possible. Reducing the network complexity either requires to reduce (or “prune”) the number of nodes or weights in the hidden layers, which means to reduce also the expressive power of the network, or to reduce the number of input nodes. This network pruning is described in details in Section 3.

For the ensemble construction of the prosody models, the author tested the W-SECA and the W-Bagging aggregation method. For W-Bagging, 6-fold cross validation was used instead of bootstrapping to obtain six different subsets of the training data. On each of these subsets, ANNs with eight different internal network structures, having one or two layers of hidden nodes and different number of nodes, were trained.

Figure 2 shows the construction of the German and the French duration models using W-Bagging aggregation. The network structures and the input factor sizes of the first eight German networks and of all French networks are given in Table 1. The best German duration ensemble has a NMSE of 0.2310 and improves the NMSE of 0.2715 of best German ANN model by about 15%. The best French duration ensemble achieves a NMSE of 0.2141, which is also an improvement of about 15% compared to the NMSE of 0.2507 Of the best French ANN model. W-SECA performed for the tested prosody corpora very similar to W-Bagging. As W-Bagging is less computational expensive than W-SECA, this method was finally applied.

3. Factor Relevance Determination

For all languages, the same initial set of input factors was used. The 910 input factors for F_0 control consisted of 190 accentuation and phrasing factors, 715 syllable structure and segmental factors, and 5 sentence length and syllable position factors. The 349 input factors for duration control consisted of 200 segmental factors, 95 accentuation, phrasing, and syllable length factors, 5 syllable level factors, 26 foot level factors, 7 phrase level factors, and 16 sentence level factors.

The optimal network structure for ANN-based prosody models is constrained by the following considerations:

- In ANN-based prosody models, the relevance of an individual input factor and the complexity of the regression problem is in general unknown. Therefore, most prosody models comprise a large number of input factors that

might be somehow relevant. However, the inclusion of many input factors has a lot of drawbacks: e.g., the interpretation of the model is more difficult, irrelevant input factors may act as input noise, the generalization capability of the model is worsened, and the demand of training samples grows exponentially with the dimensionality of the factor space. This limitation is called the “curse of dimensionality”, cf. [13].

- Neural networks having a finite number of hidden units will approximate a given function with a residual error. [14] has shown that this error decreases as the number of hidden units is increased. A traditional method for optimizing the network structure is to initially train the network with a large number of hidden nodes and later prune irrelevant weights. Because of the limited size of the prosody corpora and the large number of input factors, the number of hidden nodes of ANN-based prosodic models is in general very small. Therefore, traditional network pruning is not useful for ANN-based prosody models.

The key idea for optimizing the network structure of ANN-based prosody models is to find a trade-off between these two constraints: therefore, an initial model with a very large number of all possible input factors is trained. From this initial model, the input nodes of the least relevant factors, as determined by a factor relevance algorithm, are iteratively removed and simultaneously the number of hidden nodes is increased until the expressive power of the network optimally fits the evaluation data, cf. Figure 3.

For factor relevance determination, the author applied an extension of the so-called “optimal brain surgeon” (OBS) algorithm introduced in [15]. OBS is specially designed for pruning of ANNs and uses information from the Hessian to perform network pruning. The extension of OBS, the *Unit-OBS* algorithm, cf. [16], considers the outgoing weights of one node (unit) as a group of candidate weights: when all the weights of an unit can be deleted, the unit itself can be pruned. After pruning of an

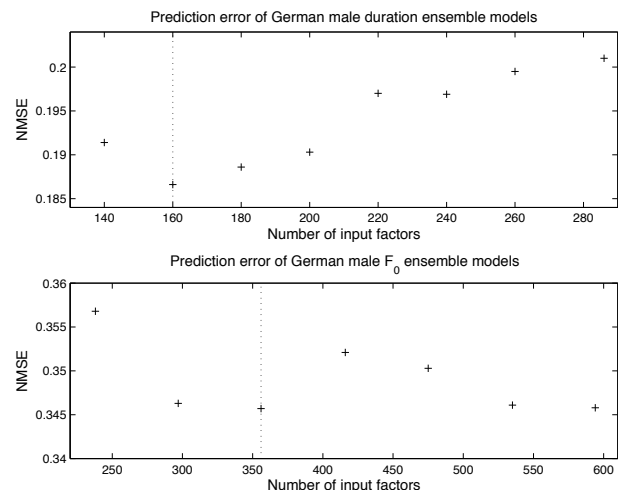


Figure 3: NMSE of duration and of F_0 prediction models for the German male prosody corpus as a function of the number of input factors. The sequence of input factors starts with the most relevant to the left. The best ensembles are indicated by a dotted line.

unit, the weights of the other units are corrected. To determine the relevance of the input factors, all input units are iteratively removed.

4. Evaluation

In [17], the performance of the neural network ensemble prosody model was evaluated on three prosody corpora: a German male, a German female, and a French female prosody corpus. Each prosody corpus has a size of about 30 minutes. The *German male corpus* consists of 186 sentences spoken by a professional male speaker in a “neutral” news-reader style. From these 186 sentences, a test set of 55 sentences was separated. All sentences were manually segmented and labeled. The *German female* and the *French female corpus* were spoken by a professional female speaker. Each corpus contains 400 sentences. From the German corpus a test set of 44 sentences and from the French corpus a test set of 48 sentences was separated. The sentences of the test sets were manually segmented and labeled, while the sentences of the training set were automatically segmented using a segmentation procedure described in [18].

Corpus	ANN	MARS	Ensemble	
German male	0.2224	0.2108	0.1866	-11.5%
German female	0.2715		0.2310	-14.9%
French female	0.2507		0.2141	-14.6%

Table 2: Normalized mean squared errors of ANN, MARS, and neural network ensemble duration models for each corpus.

Table 2 shows the NMSEs of different models for segment duration prediction. The German male ANN and MARS models were presented in [1] and were only evaluated on the German male corpus. All models apply an optimized set of most relevant input factors. The weighted neural network ensemble models perform between 14.6% to 16.1% better than the ANN models and about 11.5% better than the MARS model.

Corpus	RNN	Ensemble	
German male	0.4511	0.3457	-23.4%
German female	0.2877	0.2613	-9.2%
French female	0.3461	0.3148	-9.0%

Table 3: Normalized mean squared errors of RNN and neural network ensemble F_0 models for each corpus.

Table 3 shows the NMSEs of different models for F_0 prediction. The German male RNN model was presented in [19]. All models except the German male RNN model apply an optimized set of most relevant input factors. The weighted neural network ensemble models perform about 9% better than the RNN models with optimized sets of input factors and 23.4% better than the German male RNN model.

An additional listening experiment was conducted with 7 subjects, cf. [17]. The perceptual evaluation of 20 German and 32 French sentences, each one with its natural prosody and with the synthetic prosody predicted by the F_0 and duration models, showed that about 87.5% of the synthetic German prosody contours and about 92.5% of the synthetic French prosody contours cannot be distinguished from natural prosody contours.

5. Conclusions

Weighted neural network ensembles and factor relevance determination optimally complement one another. This is especially true in case of small training data. The neural network ensemble model outperforms standard statistical models, like CART,

ANN, or MARS, and also performs better than an ensemble model based on gradient tree boosting.

6. Acknowledgements

This work was supported by ETH Zurich and by AAP-COMET.

7. References

- [1] M. Riedi, “Controlling segmental duration in speech synthesis systems,” Ph.D. dissertation, No. 12487, Computer Engineering and Networks Laboratory, ETH Zurich (TIK-Schriftenreihe Nr. 26, ISBN 3-906469-05-0), February 1998.
- [2] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” IMS 1999 Reitz Lecture, April 2001.
- [3] P. M. Granitto, P. F. Verdes, and H. A. Ceccatto, “Neural network ensembles: Evaluation of aggregation algorithms,” *Artificial Intelligence*, vol. 163, no. 2, pp. 139–162, 2005.
- [4] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning internal representations by error propagation,” *Parallel Distributed Processing*, vol. 1, pp. 318–362, 1986.
- [5] P. J. Werbos, “Backpropagation through time: What it does and how to do it,” *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.
- [6] M. F. Møller, “A scaled conjugate gradient algorithm for fast supervised learning,” *Neural Networks*, vol. 6, no. 4, pp. 525–533, 1993.
- [7] R. E. Schapire, “The strength of weak learnability,” *Machine Learning*, vol. 5, no. 2, pp. 197–227, 1990.
- [8] H. Drucker, “Improving regressors using boosting techniques,” in *Proceedings of 14th International Conference on Machine Learning*, 1997, pp. 107–115.
- [9] L. Breiman, “Bagging predictors,” *Machine Learning*, vol. 24, no. 2, pp. 123–140, August 1996.
- [10] P. M. Granitto, P. F. Verdes, H. D. Navone, and H. A. Ceccatto, “Aggregation algorithms for neural network ensemble construction,” in *Proceedings of SBRN 2002*, 2002, pp. 178–183.
- [11] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer, 1995.
- [12] G. Rätsch, A. Demiriz, and K. P. Bennett, “Sparse regression ensembles in infinite and finite hypothesis spaces,” *Machine Learning*, vol. 48, pp. 189–218, February 2002.
- [13] R. Bellman, *Adaptive Control Processes*. Princeton University Press, 1961.
- [14] L. K. Jones, “A simple lemma on greedy approximation in hilbert space and convergence rates for projection pursuit regression and neural network training,” *Annals of Statistics*, vol. 20, no. 1, pp. 608–613, 1992.
- [15] B. Hassibi, D. G. Stork, and G. J. Wolff, “Optimal brain surgeon and general network pruning,” in *IEEE International Conference on Neural Networks*, vol. 1, San Francisco, CA, USA, March 1993, pp. 293–299.
- [16] A. Stahlberger and M. Riedmiller, “Fast network pruning and feature extraction using the unit-OBS algorithm,” in *Neural Information Processing Systems*, Denver, Colorado, USA, December 1996, pp. 655–661.
- [17] H. Romsdorfer, “Polyglot text-to-speech synthesis. Text analysis and prosody control,” Ph.D. dissertation, No. 18210, Computer Engineering and Networks Laboratory, ETH Zurich (TIK-Schriftenreihe Nr. 101), January 2009.
- [18] H. Romsdorfer and B. Pfister, “Phonetic labeling and segmentation of mixed-lingual prosody databases,” in *Proceedings of Interspeech 2005*, Lisbon, Portugal, 2005, pp. 3281–3284.
- [19] C. Traber, “SVOX: The implementation of a text-to-speech system for German,” Ph.D. dissertation, No. 11064, Computer Engineering and Networks Laboratory, ETH Zurich (TIK-Schriftenreihe Nr. 7, ISBN 3 7281 2239 4), March 1995.