DNN-BASED SPEAKER-ADAPTIVE POSTFILTERING WITH LIMITED ADAPTATION DATA FOR STATISTICAL SPEECH SYNTHESIS SYSTEMS

Miraç Göksu Öztürk,[1,3] *Okan Ulusoy,*[1,4] *Cenk Demiroglu*[2,3,4]

[1]Boğaziçi University, [2]Özyeğin University
[3]Computer Engineering, [4]Electrical & Electronics Engineering
Istanbul, Turkey

ABSTRACT

Deep neural networks (DNNs) have been successfully deployed for acoustic modelling in statistical parametric speech synthesis (SPSS) systems. Moreover, DNN-based postfilters (PF) have also been shown to outperform conventional postfilters that are widely used in SPSS systems for increasing the quality of synthesized speech. However, existing DNN-based postfilters are trained with speaker-dependent databases. Given that SPSS systems can rapidly adapt to new speakers from generic models, there is a need for DNN-based postfilters that can adapt to new speakers with minimal adaptation data. Here, we compare DNN-, RNN-, and CNN-based postfilters together with adversarial (GAN) training and cluster-based initialization (CI) for rapid adaptation. Results indicate that the feedforward (FF) DNN, together with GAN and CI, significantly outperforms the other recently proposed postfilters.

Index Terms— Speaker adaptation, speech synthesis, postfilter, deep learning

1. INTRODUCTION

Statistical parametric speech synthesis (SPSS) approach has been gaining increasing popularity in recent years in part because of the advancements made possible with the use of deep neural networks (DNNs) for acoustic modelling [1, 2, 3]. In addition to better modelling of speech parameters [1], DNNs also allow direct generation of the waveform without a parametric vocoder [4, 5].

Even though the waveform generation approach produces high-quality speech, it requires a substantial amount of data from a single speaker. Parametric methods, however, can work with only a few hours of training data. Still, they generate muffled speech because of the oversmoothing of parameter trajectories, which causes major quality degradations. To address that issue, methods such as global variance (GV) improvements and postfiltering (PF) that enhances the spectra generated with the line spectral frequency (LSF) [6] and mel-generalized cepstrum (MGC) parameters [6] have been proposed with limited success.

Recently, DNN-based postfiltering methods emerged that are shown to outperform the more conventional postfilters. In [7], two deep belief networks are cascaded with a bidirectional associative memory (BAM) and trained generatively to map synthesized spectral envelopes to natural spectral envelopes. The same DNN architecture in [7] was also trained to map synthesized mel-cepstral (MCEP) features to natural MCEP features. Even though the MCEP-based approach performed as good as the conventional methods, namely the GV and the modulation spectrum postfilter, the spectral envelope based mapping significantly outperformed them.

Similar to [7], in [8], a DNN is trained to estimate the power spectrum of natural speech from the vocoded waveform and that approach outperformed the baseline system without any postfilter. The algorithm in [7] and [8] rely on learning a mapping between vocoded and natural spectra. In [9], an auto-encoder is trained only with the natural speech signal and used as a postfilter for enhancing the vocoded speech spectrum. Thus, the postfilter in [9] is independent of the speech parameters or the parameter generation algorithm used during training. In [10], generative adversarial networks (GAN) were used to train a convolutional neural network (CNN) that generates the residual between the synthetic and natural spectral textures. A recurrent neural network (RNN)-based postfilter is proposed in [11].

All previously proposed DNN-based postfilters were trained for speaker-dependent systems where large amount of data is available for a single speaker. However, a major advantage of the SPSS approach is its ability to adapt to different speakers with limited amounts of data. For example, in [12], one-shot learning with a speech chain framework was applied for an unseen speaker to improve synthesis performance. Similarly, adaptation using speaker-embeddings [13, 14] and transfer learning methods [15] have been used for adaptation with a few seconds of data.

To retain the rapid adaptation advantage of SPSS, postfilters that perform well with few shots of data is needed.

This research is funded by TUBITAK under the project number 115E922.

Thus, here, we investigate DNN-based postfilter architectures that can be trained with a speaker-independent database and adapted to target speakers with only a few adaptation utterances. Three architectures are compared: Feedforward (FF), CNN, and RNN. Networks are trained with and without the GAN approach. Moreover, a cluster-based initialization method is used where the postfilter is initialized with a model that is trained with speakers similar to the target speaker.

2. DNN ARCHITECTURES

Our goal is to construct a speaker-adapted text-to-speech system (i.e. to generate acoustic parameters from text features) with minimal adaptation data. During model training, we first train a speaker-independent acoustic model that generates acoustic features from text features and i-vectors. Then, those features are smoothed using maximum-likelihood parameter generation (MLPG) algorithm [16]. Output of the MLPG algorithm is enhanced using a speaker-adaptive postfilters described below.

2.1. Postfilters

Let $c = [\hat{c}^T, \Delta\hat{c}^T, \Delta^2\hat{c}^T]^T$ be the output vector of the baseline network model where $\hat{c}^T, \Delta\hat{c}^T, \Delta^2\hat{c}^T$ are the cepstral, delta-cepstral and delta-delta-cepstral coefficients, respectively. Additionally, let pn and st be the 1-of-k vectors representing, respectively, the phoneme and the state information of an utterance. In the following sections, we used T, M, P, S to denote the number of frames in an utterance, the number of MGC coefficients, the size of pn vector, and the size of st vector, respectively.

Feedforward Network: The feedforward (FF) postfilter is a fully-connected model with one hidden layer having 64 units as shown in Figure 1b. Since the PF structure does not include any recurrent layer, the context information is provided to the PF model by giving the previous, c_{i-1}, and the next frame's, c_{i+1}, feature vectors together with the current feature vector c_i, where the subscript i indicates the frame number. In addition, the PF model also takes pn_i and st_i as inputs, resulting in an $(3M + P + S)$ dimensional input vector $I_i = [c_{i-1}^T, c_i^T, c_{i+1}^T, pn_i^T, st_i^T]^T$ whereas the corresponding output vector is M dimensional c_{pf_i}. There are approximately 21k trainable parameters in this model.

RNN network: We used a fully connected layer on top of a long short-term memory (LSTM) shown in Figure 1c to enhance the MGC features. A $T \times (M + P + S)$ matrix including state, phoneme and MGC features is used sequentially, one frame at a time, at the input producing an M-dimensional vector at the output for each input frame. There are approximately 39k trainable parameters in this model.

CNN network: We used the convolutional neural network (CNN) shown in Figure 1a to enhance the spectral texture of the MGC features. Batch normalization layers followed by ReLU nonlinearities are used between the convolutional layers. Different from DNN-based and RNN-based postfilters, state and phoneme features are not used as inputs. A $T \times M$ feature matrix is used at the input producing a reconstructed $T \times M$ matrix at the output. There are approximately 132k trainable parameters in this model.

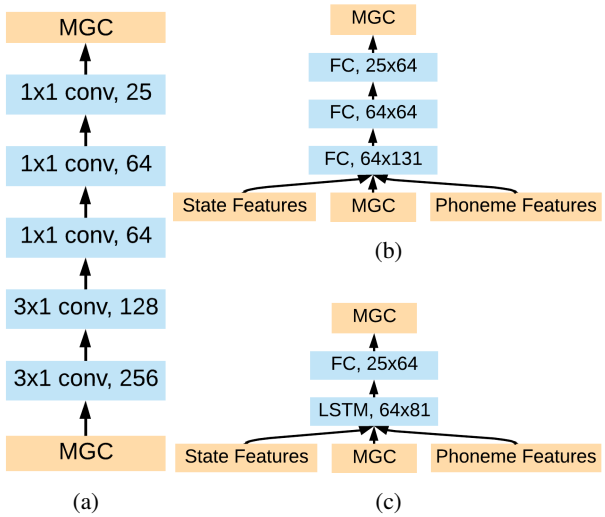

Fig. 1: (a) CNN-based postfilter, (b) Feedforward postfilter, (c) RNN-based postfilter

3. ADAPTATION

3.1. Cluster-based Initialization

Because adaptation is performed with very limited data, the optimization algorithm can quickly fall into a nearby local optima with low chances of escaping it. Thus, good initialization is important to improve the performance. We hypothesized that a PF model that is pre-adapted with the speakers in the training set whose voice characteristics are similar to a target speaker can be a better initialization for the target speaker's PF model than a more general initialization. We clustered the reference speakers into 5 groups using i-vectors with the k-means method where the number of clusters was chosen based on the initial experiments. Then, for each cluster, one model is generated by adapting the SI model with the utterances of the speakers belonging to that cluster. While adapting for a target speaker, the model is initialized with one of these 5 pre-trained models that is closest to the target speaker. Euclidean distance of each cluster's mean vector to the target speaker's i-vector is used as the selection criterion.

3.2. Adversarial Training

When GAN is not used, the PF is trained using the standard Mean Squared Error (MSE) loss as below:

$$L_{MSE}(\hat{y}, y) = \frac{1}{T}\sum_{t=0}^{T}(y_i - \hat{y}_i)^2 \quad (1)$$

where $\hat{y}_i$ is a vector prediction for the time step i and y_i vector is the target parameters for the same time step. In the adversarial approach, in addition to the MSE loss, a binary cross entropy loss function

$$L_{BCE}(\hat{\mathrm{y}}, \mathrm{y}) = \frac{1}{T}\sum_{t=0}^{T} -(\mathrm{y}_i log(\hat{\mathrm{y}}_i) + (1-\mathrm{y}_i) log(1-\hat{\mathrm{y}}_i)) \quad (2)$$

is used where $\hat{\mathrm{y}}_i$ is a scalar prediction for the time step i and $\hat{\mathrm{y}}_i \in \{x \in R \mid 0 < x < 1\}$ while y_i is the target value for the time step i and $\mathrm{y}_i \in (0, 1)$ denoting the label as either fake(prediction) or real(natural).

The loss function with the adverserial approach is

$$\mathcal{L}_{PF} = L_{MSE}(c_{pf}, c_{nat}) + \frac{E_{L_{MSE}}}{E_{L_{BCE}}} L_{BCE}(ap_{cpf}, \mathbf{1}) \quad (3)$$

where $E_{L_{MSE}}$ and $E_{L_{BCE}}$ are the expected values of MSE and binary cross entropy losses. ap_{cpf} is the prediction of the adversarial network when its input is the parameters generated by the PF network and $\mathbf{1}$ is a vector of ones that has the same length as ap_{cpf}. Thus, the loss function increasingly penalizes c_{pf} if it diverges from the natural parameters c_{nat} while trying to minimize the MSE.

$E_{L_{MSE}}/E_{L_{BCE}}$ can be considered as the scaling factor to boost the effect of L_{BCE} since $E_{L_{MSE}} >> E_{L_{BCE}}$ and the gradients from L_{MSE} and L_{BCE} differ significantly in magnitude.

Adversarial network training uses the following loss functions:

$$\mathcal{L}^1_{ADV} = L_{BCE}(ap_{cpf}, \mathbf{0}) \quad (4)$$

$$\mathcal{L}^2_{ADV} = L_{BCE}(ap_{cnat}, \mathbf{1}) \quad (5)$$

where ap_{cpf} and ap_{cnat} are the predictions for the generated parameters of the PF model and the natural parameters respectively. $\mathbf{0}$ and $\mathbf{1}$ are the vectors of zero and one. The decision to pick either $\mathcal{L}^1_{ADV}$ or $\mathcal{L}^2_{ADV}$ for the weight update is made in a balanced and mutually exclusive manner as carried out typically in the training process of a GAN model's discriminator component.

4. EXPERIMENTS

4.1. Experimental setup

All experiments were conducted on the Wall Street Journal (WSJ) speech database. 154 features including 25 Mel-Generalized Cepstrum Coefficients (MGC), 1 log of fundamental frequency (LF0) and 25 Band Aperiodicity (BAP) together with their delta and delta-delta features were extracted from speech data at a sampling rate of 16 KHz and 5 msec frame rate. Additionally, a voiced/unvoiced binary feature was appended to represent the voicing information.

A 5-state HMM model was applied to model and align phonemes by using HTS 2.3 speech synthesis tool [17]. Among the total of 156 speakers, 135 of them were used for training, whereas the test and adaptation processes were performed on the remaining 21 target speakers' data. Adaptation was performed with 5, 10, and 15 seconds of data.

The acoustic model is a Deep Neural Network (DNN) model with three 512-node feed-forward (FF) layers followed by one 256-node Long-Short Term Memory (LSTM) layer and one 154-node FF output layer. The model is trained with approximately 5 hours of balanced speech data taken from 135 male speakers. The DNN model is trained for 50 epochs and Adam optimizer [18]. For speaker adaptation, i-vectors [19] augmented with the text features are used as input to the DNN.

In listening tests, AB test is used for comparison of speech quality whereas ABX test is used for comparison of speaker similarity. [1]

Table 1: Mel cepstral distortion (MCD) scores of the speaker-independent postfilters with and without cluster-based initialization (CI) are shown. Scores for tandem use of FF- and RNN-based postfilters with the CNN-based postfilter are also shown.

POSTFILTER	MCD
SI-Baseline	5.19
FF-SI-PF	5.89
RNN-SI-PF	**5.16**
CNN-SI-PF	5.45
FF-SI-PF-CI	5.60
RNN-SI-PF-CI	5.23
CNN-SI-PF-CI	5.47
FF-SI-PF on CNN-SI-PF	6.15
RNN-SI-PF on CNN-SI-PF	5.34

4.2. Results and Discussion

Performance of the Speaker-Independent Postfilters: MCD scores of all postfilters without any adaptation are shown in Table 1. RNN-based postfilter performed the best among all three postfilters. Cluster-based initialization (CI) of the network, without any adaptation, did not significantly affect the performance of RNN and CNN. However, the FF system is significantly improved with the CI approach. Because the FF system can adapt with significantly less data than the RNN and CNN methods, as discussed in more detail below, the CI approach was successful with the FF-based postfilter but not with the others. Still, FF-based postfilter performed worse than the other two postfilters in the AB and ABX tests with and without CI.

Both RNN and CNN postfilters significantly improved speech quality in the listening tests as shown in Figure 2.

[1] Audio samples can be found at https://mgoksu.github.io/icassp19/index.html

Performance-wise, two methods were not found to be significantly different as shown in the same figure. In our listening tests, we also found that the perceived speaker similarity also improved with those two postfilters, even though no speaker adaptation was performed, thanks to the improvement in quality with postfiltering.

Table 2: Mel cepstral distortion (MCD) scores of the speaker-adapted postfilters with 5, 10, and 15 seconds of adaptation data.

POSTFILTER	5 sec	10 sec	15 sec
FF	5.74	5.68	5.65
FF+CI	5.45	5.35	5.31
FF+CI+ADV	5.40	5.16	5.11
FF on CNN-PF	5.69	5.60	5.52
RNN	**5.15**	**5.14**	5.15
RNN+CI	5.21	5.20	5.20
RNN+CI+ADV	5.35	5.15	**5.07**
RNN on CNN-PF	5.30	5.31	5.31
CNN	5.45	5.31	5.27
CNN+CI	5.45	5.29	5.25
CNN+CI+ADV	7.24	7.01	6.99

Performance of the Speaker-Adapted Postfilters: The CNN-based system could not adapt to the target speakers even when 15 seconds of adaptation data was used as shown in Table 2. We found that not only the speech quality, but also the speaker similarity significantly degraded after adaptation. Because the CNN system is learning to map large spectral textures, attempting to adapt it with only a few shots of data had a detrimental effect because of the large number of parameters that need to be adapted. Still, because it performs as well as the RNN postfilter for the SI case, we attempted to use it in tandem with the RNN-based and FF-based postfilters. However, that approach slighly degraded the performance of the two postfilters as shown in Table 2.

Similar to CNN, RNN-based postfilter could not adapt to new speakers with limited data. However, performance did not degrade with adaptation. FF-based postfilter was the most effective at adaptation as shown in Table 2.

Cluster-based initialization significantly improved the adaptation capability of the FF postfilter whereas it slightly degraded the performance of the RNN-based postfilter. Both RNN and RNN+CI system performances remain almost constantly for all adaptation data sizes. This is because RNN system cannot adapt with limited data regardless of the initialization.

Adverserial training improved the performance of the FF-based postfilter whereas it degraded the performance of the CNN-based postfilter as shown in Table 2. Similarly, adverserial training degraded the RNN-based postfilter except for the 15sec case.

ABX speaker similarity test results are shown in Figure 3

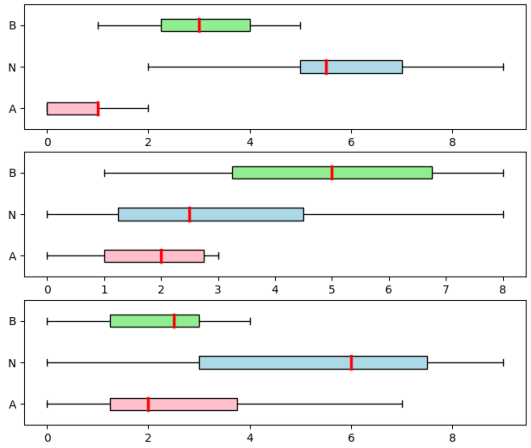

Fig. 2: In the top figure, SI-baseline system (A) is compared with the RNN-SI postfilter (B) using the AB test. Significance (p-value) is 0.01. In the middle figure, SI-baseline system (A) is compared with the RNN-SI postfilter (B) using the AB test. Significance (p-value) is 0.01. In the bottom figure, RNN-SI posttfilter (A) is compared with the CNN-SI postfilter (B) using the AB test. Significance (p-value) is 0.55.

for the comparison of RNN and FF+CI+ADV postfilters. FF+CI+ADV postfilter clearly outperformed the RNN-based postfilter in all cases. Similar results were obtained for the AB quality tests even though they are not presented here due to space constraints. Thus, we conclude that in a speaker-adaptive postfilter setting with a few seconds of adaptation data, the FF postfilter together with CI and adverserial training significantly outperforms the CNN- and RNN-based postfilters.

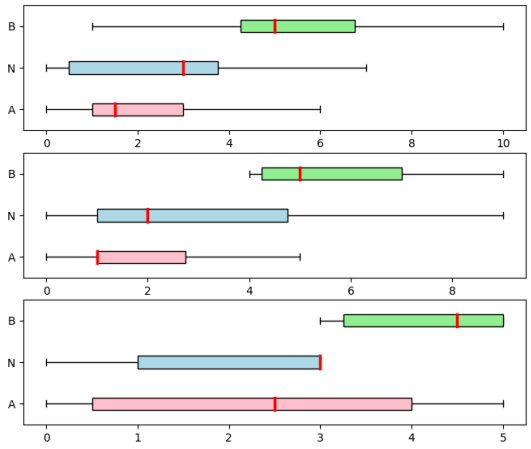

Fig. 3: RNN postfilter (A) is compared with the FF+CI+ADV postfilter (B) using the ABX test. Results are shown when the adaptation data is 5 sec (top figure), 10 sec (middle figure), 15 sec (bottom figure). Significance (p-value) of the 5 and 10 sec cases are 0.01 whereas significance of the 15 sec case is 0.03.

5. REFERENCES

[1] A. Senior H. Zen and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," *Proc. ICASSP, 2013, pp. 79627966*, 2013.

[2] Yao Qian, Yuchen Fan, Wenping Hu, and Frank K Soong, "On the training aspects of deep neural network (dnn) for parametric tts synthesis," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 3829–3833.

[3] Heiga Zen and Andrew Senior, "Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2014, pp. 3872–3876.

[4] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio," pp. 1–15, 2016.

[5] Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis C Cobo, Florian Stimberg, Norman Casagrande, Dominik Grewe, Seb Noury, Sander Dieleman, Erich Elsen, Nal Kalchbrenner, Heiga Zen, Alex Graves, Helen King, Tom Walters, Dan Belov, and Demis Hassabis, "Parallel wavenet: Fast high-fidelity speech synthesis," *arXiv preprint arXiv:1711.10433*, 2017.

[6] Keiichi Tokuda, Takao Kobayashi, Takashi Masuko, and Satoshi Imai, "Mel-generalized cepstral analysis-a unified approach to speech spectral estimation," in *Third International Conference on Spoken Language Processing*, 1994.

[7] Ling-Hui Chen, Tuomo Raitio, Cassia Valentini-Botinhao, Zhen-Hua Ling, and Junichi Yamagishi, "A deep generative architecture for postfiltering in statistical parametric speech synthesis," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 11, pp. 2003–2014, 2015.

[8] Takuma Okamoto, Kentaro Tachibana, Tomoki Toda, Yoshinori Shiga, and Hisashi Kawai, "Deep neural network-based power spectrum reconstruction to improve quality of vocoded speech with limited acoustic parameters," *Acoustical Science and Technology*, vol. 39, no. 2, pp. 163–166, 2018.

[9] Ya-Jun Hu, Zhen-Hua Ling, and Li-Rong Dai, "Deep belief network-based post-filtering for statistical parametric speech synthesis," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5510–5514.

[10] Takuhiro Kaneko, Hirokazu Kameoka, Nobukatsu Hojo, Yusuke Ijima, Kaoru Hiramatsu, and Kunio Kashino, "Generative adversarial network-based postfilter for statistical parametric speech synthesis," in *Proc. ICASSP*, 2017, vol. 2017, pp. 4910–4914.

[11] Prasanna Kumar Muthukumar and Alan W Black, "Recurrent neural network postfilters for statistical parametric speech synthesis," *arXiv preprint arXiv:1601.07215*, 2016.

[12] Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura, "Machine speech chain with one-shot speaker adaptation," *arXiv preprint arXiv:1803.10525*, 2018.

[13] Songxiang Liu, Jinghua Zhong, Lifa Sun, Xixin Wu, Xunying Liu, and Helen Meng, "Voice conversion across arbitrary speakers based on a single target-speaker utterance," *Proc. Interspeech 2018*, pp. 496–500, 2018.

[14] Ju-chieh Chou, Cheng-chieh Yeh, Hung-yi Lee, and Lin-shan Lee, "Multi-target voice conversion without parallel data by adversarially learning disentangled audio representations," *arXiv preprint arXiv:1804.02812*, 2018.

[15] Sercan O Arik, Jitong Chen, Kainan Peng, Wei Ping, and Yanqi Zhou, "Neural voice cloning with a few samples," *arXiv preprint arXiv:1802.06006*, 2018.

[16] Keiichi Tokuda, Takao Kobayashi, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura, "Speech parameter generation algorithms for hmm-based speech synthesis," in *Proc. ICASSP*, 2000, pp. 1315–1318.

[17] "Hmm-based speech synthesis system (hts), available at: http://hts.sp.nitech.ac.jp/," .

[18] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[19] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.