A SPECTRALLY WEIGHTED MIXTURE OF LEAST SQUARE ERROR AND WASSERSTEIN DISCRIMINATOR LOSS FOR GENERATIVE SPSS

Gilles Degottex[1,2]*, Mark Gales*[2]

[1]ObEN, Inc., Pasadena, US
[2]University of Cambridge, UK

ABSTRACT

Generative networks can create an artificial spectrum based on its conditional distribution estimate instead of predicting only the mean value, as the Least Square (LS) solution does. This is promising since the LS predictor is known to oversmooth features leading to muffling effects. However, modeling a whole distribution instead of a single mean value requires more data and thus also more computational resources. With only one hour of recording, as often used with LS approaches, the resulting spectrum is noisy and sounds full of artifacts. In this paper, we suggest a new loss function, by mixing the LS error and the loss of a discriminator trained with Wasserstein GAN, while weighting this mixture differently through the frequency domain. Using listening tests, we show that, using this mixed loss, the generated spectrum is smooth enough to obtain a decent perceived quality. While making our source code available online, we also hope to make generative networks more accessible with lower the necessary resources.

***Index Terms*—** Text-to-speech, acoustic model, convolutive networks, generative adversarial networks.

1. INTRODUCTION

Deep Neural Nets (DNN) have enabled a large improvement in the quality of Statistical Parametric Speech Synthesis (SPSS) [1, 2] compared to previous approaches with explicit distribution modeling [3]. Using Least Square (LS) training, a speaker-dependent DNN model can be built on only one hour of data and trained within a few hours or even minutes with modern hardware. This allows a rapid development cycle, as well as systems for new voices. The main downside of the LS training is that it basically predicts the mean value of the conditional distribution, which leads to the well known averaging, oversmoothing and muffling effects.

Generative Adversarial Networks (GAN) [4, 5] are potential solution to this oversmoothing issue by generating samples according to the whole underlying distribution instead of the mean value only. However, modeling a whole distribution requires more data than that necessary to predict a reliable mean estimate (similarly to a noisy histogram computed from a too sparse sampling). Consequently, GAN-based training, or similarly WaveNet-based approaches [6], require usually more data for training than LS training. In SPSS, if a too small amount of data is used, the resulting quality is degraded compared to LS, the voice is often hoarse and surrounded by many artifacts.

In this paper, using $\approx$1 hour of data for a single speaker model, we suggest a combination of a least square and WGAN discriminator losses [5] for training the generator. In doing so, the aim is to balance the artifact-free mean prediction and the rich, but noisy, generative approach. This is similar to [7] where the Minimum Generation Error is used instead of the frame-based LS error. In our study, we also use non-cepstral features as in [8], which simplifies the overall process and the interpretation of the system performance. Through visual inspection of the amplitude spectrum generation (See e.g. Fig.2), we can see that the details in the high frequencies are the most oversmoothed by LS training. This is mainly due to the frequency warping of the mel-scale used for increasing the frequency resolution in the low frequencies. Therefore, we also suggest the addition of a spectral weight to the LS and WGAN loss function. We give more weight to the WGAN discriminator loss on the higher frequencies and ensure the smooth reconstruction of the low frequencies by forcing the LS loss in this frequency band. The final loss mixture will be called WLSWGAN in the following.

By appropriately weighting the loss funcion, this WLSWGAN approach should lead to improved quality compared to the traditional WGAN loss, while only requiring a limited amount of data. Avoiding the need for large synthesis training corpora is an interesting aspect for text-to-speech synthesis. Smaller corpora are both easier to build and cheaper. Additionally, many small corpora are already available and it also limit the need of computational power. All experiments in this paper have been run on a single modern GPU (NVIDIA GTX 1080 TI or similar (e.g. Tesla P100)) in $\approx$1 day. This leads to a very efficient SPSS pipeline that has been made publicly available [1] under the name *Percival*.

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 655764.

[1]https://gitlab.com/gillesdegottex/percivaltts

The next section describes the the mixture of LS loss and WGAN discriminator loss. Section 3 describes the default architecture used in *Percival* as well as other practical elements. The evaluation section present results of listening tests using various training setups to evaluate the improvement obtained with WLSWGAN.

2. MIXED LOSSES OF WEIGHTED LEAST SQUARE + WASSERSTEIN GAN (WLSWGAN) DISCRIMINATOR

Because we use a vocoded speech signal in this work, we generate spectral features (fundamental frequency (f_0), amplitude spectral envelope and noise-related feature). Conversely to many previous works using cepstral representation for the amplitude spectral envelope, we chose here to use a simple log spectral representation [8] which are only warped through frequency using a mel scale. For reason of clarity we will now focus on the generation of the amplitude spectral envelope. The next section will describe the overall architecture.

For a training set $\left\{\{\boldsymbol{x}^{(i)}, \boldsymbol{z}^{(i)}, \boldsymbol{y}^{(i)}\}\right\}_{i=1:n}$ of contextual inputs $\boldsymbol{x}^{(i)}$, noise vectors $\boldsymbol{z}^{(i)}$ (in $[-1, 1]$, of size 100 in this work) and spectral feature vectors $\boldsymbol{y}^{(i)}$, the usual loss function of the generator using LS at a given frame is:

$$\mathcal{L}_{\mathtt{g}}(\boldsymbol{\theta}_{\mathtt{g}}) = \sum_{i=1}^{n} ||\boldsymbol{y}^{(i)} - \mathcal{G}(\boldsymbol{x}^{(i)}, \boldsymbol{z}^{(i)}; \boldsymbol{\theta}_{\mathtt{g}})||^2 \tag{1}$$

where $\boldsymbol{\theta}_{\mathtt{g}}$ are the generator parameters and the training process minimizes $\mathcal{L}_{\mathtt{g}}(\boldsymbol{\theta}_{\mathtt{g}})$ over $\boldsymbol{\theta}_{\mathtt{g}}$. With LS, $\boldsymbol{z}$ has no influence on the result since the LS forces $\mathcal{G}(.)$ to be deterministic. We used it in (1) for consistency with the following.

With WGAN [5], the usual loss function for the generator would be:

$$\mathcal{L}_{\mathtt{g}}(\boldsymbol{\theta}_{\mathtt{g}}|\boldsymbol{\theta}_{\mathtt{d}}) = -\sum_{i=1}^{n} \mathcal{D}(\mathcal{G}(\boldsymbol{x}^{(i)}, \boldsymbol{z}^{(i)}; \boldsymbol{\theta}_{\mathtt{g}}); \boldsymbol{\theta}_{\mathtt{d}}) \tag{2}$$

where $\mathcal{D}(\tilde{\boldsymbol{y}}; \boldsymbol{\theta}_{\mathtt{d}})$ usually integrates over the frequency dimension using a simple Fully Connected (FC) layer and $\mathcal{L}_{\mathtt{g}}(\boldsymbol{\theta}_{\mathtt{g}}|\boldsymbol{\theta}_{\mathtt{d}})$ is minimized over $\boldsymbol{\theta}_{\mathtt{g}}$ during training, while keeping $\boldsymbol{\theta}_{\mathtt{d}}$ frozen. The loss for the discriminator is:

$$\mathcal{L}_{\mathtt{d}}(\boldsymbol{\theta}_{\mathtt{d}}|\boldsymbol{\theta}_{\mathtt{g}}) = -\sum_{i=1}^{n} \Big(\mathcal{D}(\boldsymbol{y}^{(i)}; \boldsymbol{\theta}_{\mathtt{d}}) - \mathcal{D}(\mathcal{G}(\boldsymbol{x}^{(i)}, \boldsymbol{z}^{(i)}; \boldsymbol{\theta}_{\mathtt{g}}); \boldsymbol{\theta}_{\mathtt{d}})\Big) \tag{3}$$

where $\mathcal{L}_{\mathtt{d}}(\boldsymbol{\theta}_{\mathtt{d}}|\boldsymbol{\theta}_{\mathtt{g}})$ is minimized over $\boldsymbol{\theta}_{\mathtt{d}}$, while keeping $\boldsymbol{\theta}_{\mathtt{g}}$ frozen. During training, $\mathcal{L}_{\mathtt{d}}$ and $\mathcal{L}_{\mathtt{g}}$ are minimized alternatively with an iteration ratio of 5:1, respectively. As argued in the introduction, even though the WGAN training offers a sampling of the underlying distribution instead of the mean as in the LS solution, the samples lack a constraint that ensures a minimum time-frequency consistency when the quantity of data is too sparse with respect to the complexity of its distribution. To ensure this smoothness, we suggest to mix the LS term with the original WGAN discriminator loss for training the generator:

$$\begin{aligned}\mathcal{L}_{\mathtt{g}}(\boldsymbol{\theta}_{\mathtt{g}}|\boldsymbol{\theta}_{\mathtt{d}}) = \sum_{i=1}^{n} &- \mathcal{D}(\mathcal{G}(\boldsymbol{x}^{(i)}, \boldsymbol{z}^{(i)}; \boldsymbol{\theta}_{\mathtt{g}}); \boldsymbol{\theta}_{\mathtt{d}}) \\ &+ ||\boldsymbol{y}^{(i)} - \mathcal{G}(\boldsymbol{x}^{(i)}, \boldsymbol{z}^{(i)}; \boldsymbol{\theta}_{\mathtt{g}})||^2\end{aligned} \tag{4}$$

Additionally, we assume that the low frequencies have to be the most constrained whereas the higher frequencies can be left guided mainly by the WGAN loss. We use the subscript k for the kth bin of the spectral representation and (4) can thus be extended with respect to k:

$$\begin{aligned}\mathcal{L}_{\mathtt{g}}(\boldsymbol{\theta}_{\mathtt{g}}|\boldsymbol{\theta}_{\mathtt{d}}) = \sum_{i=1}^{n} &- \mathcal{D}(\mathcal{G}(\boldsymbol{x}^{(i)}, \boldsymbol{z}^{(i)}; \boldsymbol{\theta}_{\mathtt{g}}); \boldsymbol{\theta}_{\mathtt{d}}) \\ &+ \frac{1}{K} \sum_{k=0}^{K-1} w_k \Big(\boldsymbol{y}_k^{(i)} - \mathcal{G}_k(\boldsymbol{x}^{(i)}, \boldsymbol{z}^{(i)}; \boldsymbol{\theta}_{\mathtt{g}})\Big)^2\end{aligned} \tag{5}$$

where K is the number of frequency bin from DC to Nyquist frequencies and w_k is the weighting term. In this work, a sigmoid function $\sigma(.)$ is used to determine w_k:

$$w_k = 1 - (1 - \alpha) \cdot \sigma(-(k_{\mathtt{c}} - k) \cdot k_s) \tag{6}$$

below a cut-off dimension $k_{\mathtt{c}}$, w_k tends to one (maximum weight for LS) and, above, it tends to α. Through trial-and-error experiments, we chose $k_{\mathtt{c}}$ so that the center of the sigmoid is at 4kHz, $k_s = 1/8$ and $\alpha = 0.25$. This weighting is suggested here only as an example and for the purpose of the experiments in Sec. 4. A thorough study of this parameterization might be necessary. Thus, the LS always weights the most in the low frequencies and weights only 0.25 in the mid and high frequencies. Note that these two parameters are kept the same for all the experiments of Sec. 4. Future works could design and investigate better weighting functions. Since the LS drives most of the loss in the low frequencies and the WGAN cannot work on this part of the spectrum, it is also important to weight the discriminator input accordingly. Otherwise, the discriminator will be fed by an error that it cannot reduce. The input to the discriminator is thus weighted by $1 - w(k) = (1 - \alpha) \cdot \sigma(-(k_c - k) \cdot k_s)$

Note that the WGAN loss is in $(-\infty; +\infty)$ whereas the LS is in $[0, +\infty)$. Mixing these two losses might thus seem a bit surprising. However, as long as the WGAN component does not lead to meaningful solution, the LS loss will be substantial. During training, as soon as the LS becomes small enough, the WGAN loss becomes comparatively more important and can drive the training. In other words, the mix forces the solution to go first close enough to the mean, where the WGAN can then start correcting the solution towards a more complex sampling scheme. Mixing these two losses might also be a way to improve the first initialization epochs of the training. A future work might focus on decreasing $w(k)$ through the epochs.

3. DEEP CONVOLUTIVE NEURAL NETWORK (DCNN) ARCHITECTURE

The default architecture of the generator in *Percival* is a Deep Convolutional Neural Network (DCNN) (see top part of Fig. 1). The context labels are first pre-processed by a 1-dimensional CNN of 100ms in order to build 4 feature maps, which are then followed by 2 stacked Fully Connected (FC) layers of 256 units. This contextual information is used as input for each of the three vocoder features (f_0, the frequency warped spectrum, and the noise-related feature (noise mask for PML[9, 10] or the aperiodicity for the more traditional WORLD/STRAIGHT vocoders[11]). The f_0 is modeled by a BLSTM [12] layer of 256 units. The amplitude spectral envelope is modeled by 8 stacked Gated 2-dimensional convolutional (GCNN) layers with 16 filters of dimension 5x5. For the noise component 4 FC layers of 256 units were used. This simple architecture was chosen for the noise component because no obvious improvement were obtained using DCNN.

The default architecture for the discriminator first ignores the f_0 and noise features in order to train only the amplitude spectrum with the WLSWGAN loss. The f_0 modeling often suffers from oversmoothing and exploiting the GAN approach would be an interesting element to study. In practice, we could not obtain satisfactory results when trying to optimize a BLSTM with WGAN and replacing the BLSTM with a DCNN was not leading to any better result than optimizing the BLSTM with LS. The reason to ignore the noise feature in the discriminator is similar. The WGAN optimized noise tends to adds spurious artefacts. The architecture of the discriminator for the amplitude spectrum follows a reversed process compared to the generator (see bottom part of Fig. 1). The spectrum is first analyzed by a 8 stacked GCNN with 16 filters of dimension 5x5. The result is then concatenated with that of the context labels pre-processor. Finally, 6 stacked FC layers of 256 units finish the discriminator with a very final FC layer of 1 unit for the output.

3.1. Implementation

The *Percival*[2] pipeline is originally implemented in Theano / Lasagne[3]. It offers a few features: Possibility to recover training states; Generalization of the vocoder used (in addition to PML, WORLD[13] is also available); Smoke tests for ensuring that features are always working through continuous integration; The trainings are also repeatable, which eases debugging.

The data are provided through minibatches of 5 sentences of variable time duration. In practice the time duration is capped to 2s to avoid memory overflow. For each iteration, a random segment of 2s is selected from 5 sentences picked randomly. These segments are stacked to obtain a batch of dimension (5,400,F). Where 400 corresponds to 2 seconds sampled each 5ms and F is the concatenation size of the acoustic features. The same segments are picked from the context labels to build the corresponding input batch.

There is no normalization scheme in the discriminator. Indeed, batch normalization should not be used [5]. We tried local response normalization[14], that improved some aspects of the voice, but clearly increased the quantity of artifact. Some other normalization schemes should be investigated (e.g. layer normalization).

4. EVALUATION

In this section, we present the results of listening tests using assessment of overall perceived quality of different systems. We trained the two types of models for 10 different voices, 5 males and 5 females, all American or British English (SLT16(female)[15], LS(f)[16], BDL32(male)[15], RMS(m)[15], Nick(m) [17], as well as 3 female voices FC, FI, FL and 2 male voices MC, MN from ObEN, Inc., see list below in Table 1). For each voice, the data is split in 3 sets. The last 50 sentences are used for the listening tests, the 50 previous sentences are used for validation during training and the rest is used as training data.

We used the PML[9] vocoder for its good quality compared to other vocoders [10]. A log scale is used for the continuous f_0 values (there is no voicing decision in PML; we used REAPER [18] for the f_0 estimate). A log scale is also used for the warped amplitude spectrum (using WORLD's estimate[13]), which is sampled with 129 bins. For the noise, PML's noise mask[9], a binary mask in time and frequency, is also warped using a mel scale and sampled using 33 bins. This exact same feature setup is used for the 10 voices. There is no extra deltas concatenated to this feature set and the MLPG[19] is thus not needed at synthesis.

Since the WGAN-based training is supposed to remove muffling effects eventually, it could be interesting to study the need of Post-Processing (PP) techniques. However, from internal experiments, by applying Merlin's post-processing[2], the quality of all methods is generally improved, for both LS and WLSWGAN optimization. Comparing, e.g. BLSTM+LS with PP vs. DCNN+WLSWGAN without PP, is thus not interesting since any method with PP would be favored. Keeping in mind that the quality of the results presented in this section can be even further improved using PP, we preferred to keep it aside in these experiments for reason of simplicity and thus focus on the raw generation of the acoustic features. Note that this PP option is available in *Percival* through a simple flag at synthesis stage.

Concerning the text and durations, the Merlin[2] pipeline is used to create aligned contextual labels. For synthesis, the durations used were that of the original aligned context labels, since the focus of this work is on acoustic modeling only.

Concerning the training process, hyper parameters (e.g.

[2] https://gitlab.com/gillesdegottex/percivaltts

[3] It has now been refactored to TensorFlow/Keras under release 2.0

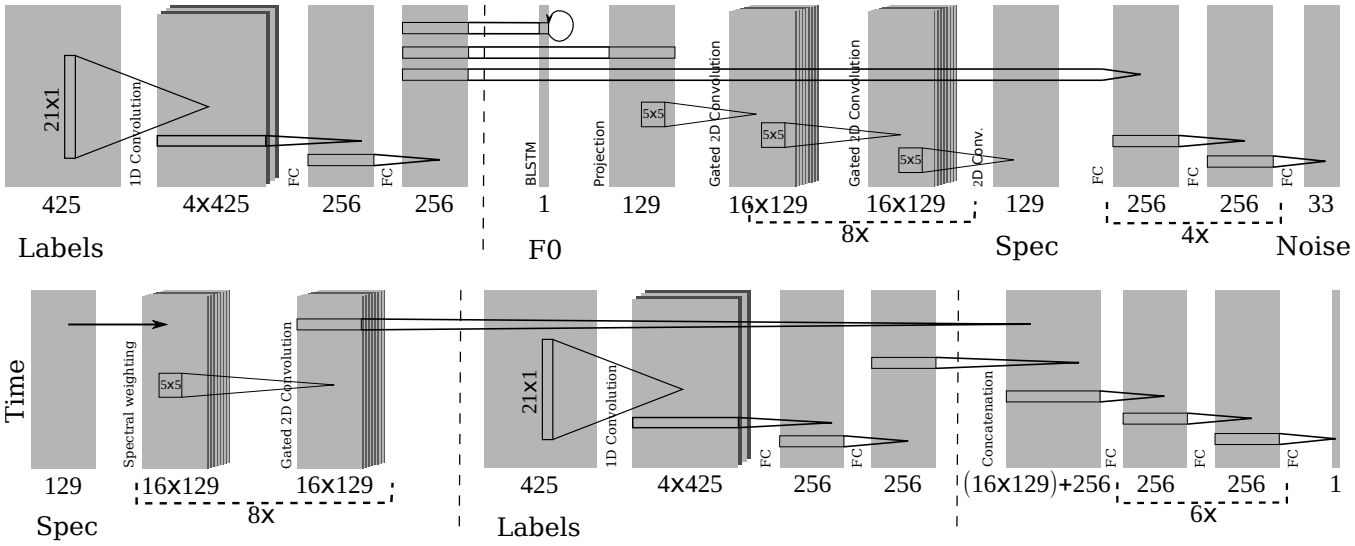

Fig. 1. Architecture of the generator (top) and discriminator (bottom)

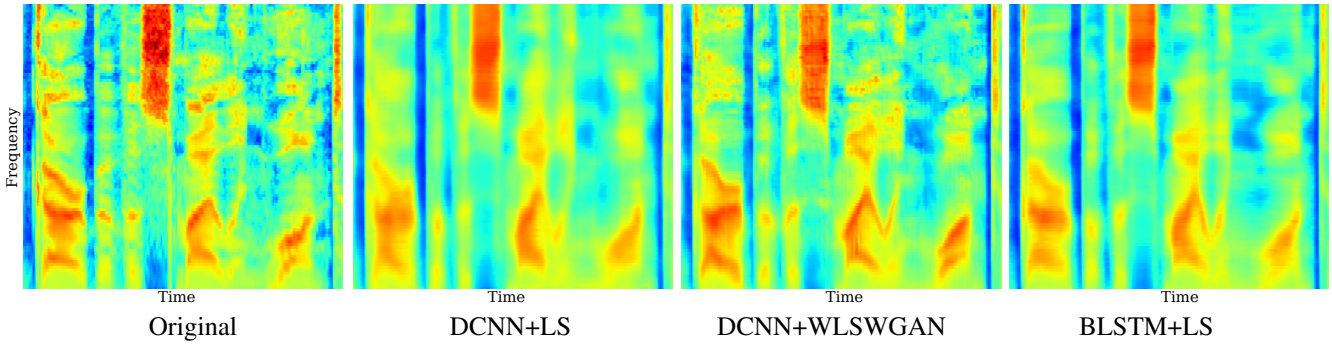

Fig. 2. Example of frequency warped amplitude spectrogram generated by WLSWGAN or LS. The DCNN-WLSWGAN generation is indeed closer to the original than the DCNN+LS and BLSTM+LS generation. More details are obtained in the higher frequencies and the formant's shape in the lower frequencies are clearer.

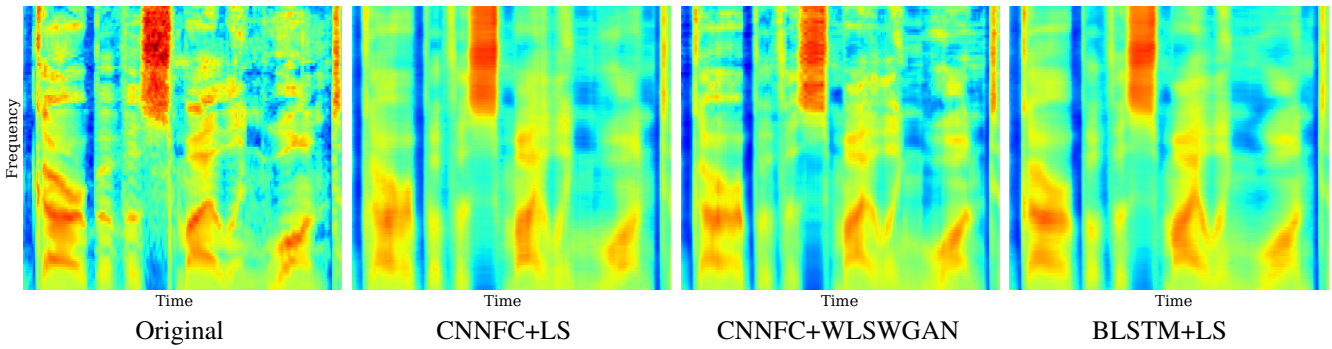

Fig. 3. Example of frequency warped amplitude spectrogram generated/predicted by CNNFC model and optimized using LS or the suggested WLSWGAN. The WGAN-based generation is indeed closer to the original than the LS-based prediction.

batch size, learning rates.) are not optimized. The default values present in *Percival* are used for all of the 10 voices. On the one hand, this might have limit the overall quality of the syntheses. On the other hand, this also shows that the *Percival* pipeline can already provide a decent *out-of-the-box* quality.

As baseline system, we also trained 3-stacked BLSTM model of 256 units using LS loss, which has been proven to be a reliable model offering a good quality (sounds are available online[4]). We have not been able to obtain any decent result by training any RNN-based network (e.g. BLSTM) us-

Name	*gender*	f_s [kHz]	*train data* [h:m:s]
SLT16	female	16	00:50:38
LS	female	32	02:13:54
BDL32	male	32	00:50:32
RMS	male	16	00:59:12
Nick	male	48	01:49:32
FC	female	48	00:53:54
FI	female	48	00:48:57
FL	female	48	01:08:08
MC	male	48	00:55:33
MN	male	48	01:11:08

Table 1. Characteristics of the training data for each voice.

ing WGAN or WLSWGAN. Most of these trainings either did not converge to anything meaningful, or resulted in many artifacts and musical sounds.

We also compared the generations with PML's resynthesis. Even though the original recording is often used in other studies, we believe that the resynthesis is more appropriate for SPSS evaluation since its quality is the absolute upper bound in these experiments. There are no other common hidden anchor among the system compared (e.g. a deliberately degraded signal as for MUSHRA tests [20]) This means that the results are not comparable to any other listening tests results. This also prevent the listeners to artificially compress the results towards a high quality by blaming systematically the same type of sounds. It forces them to focus on more subtle differences and use more of the grading scale available. To verify the absolute quality, the reader can listen to online sound samples[4].

4.1. Deep Convolutive Neural Network (DCNN)

In this first experiment, we evaluate the quality resulting from standard LS training, standard WGAN training and the suggested WLSWGAN training using the model of Fig. 1. Examples of generated amplitude spectral envelopes are shown in Fig.2 (the DCNN + original WGAN is not present as the spectrogram is very poor). A Mean Opinion Score (MOS)[21] listening test has been carried out to ask listeners to assess the overall perceived quality of syntheses generated using the systems BLSTM+LS, DCNN+LS, DCNN+WGAN, DCNN+WLSWGAN and the quality of PML's resynthesis, for only one sentence of each voice picked up randomly among the test set of 50 sentences. Using crowdsourcing, workers from Amazon Mechanical Turk were asked to take the test [22, 23]. 33 listeners took the test properly and Fig. 4 shows the aggregated results for all the voices (detailed results are in Fig. 6.

From Fig. 4, we can see that the WLSWGAN training of the DCNN-based architecture does indeed provide a way better quality than using the standard WGAN (with p-value$<$0.01 on the means' difference). This confirms the main claim of this paper. The quality is also better than that of

[4] http://gillesdegottex.eu/Demos/Percival2018

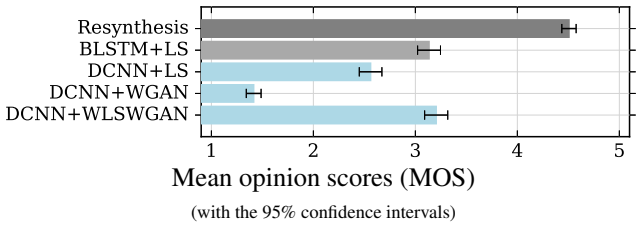

Fig. 4. MOS about perceived quality of various models using: LS, WGAN or the suggested WLSWGAN optimization.

the traditional LS training (DCNN+LS). However, there is no difference compared to the baseline BLSTM model trained with LS. By inspection of the sounds[4], this last observation might seem surprising. It seems the listeners will always favor the smoothest stimuli, even against a more realistic, though slightly noisier, stimuli.

4.2. Light architecture

In this second experiment we show that the WLSWGAN training approach is also effective on a light architecture. For this purpose, the model is simply the context label pre-processor (first part on the left of the generator in Fig. 1) that we complete with 3 extra FC layers of 256 units. This light model will be called CNNFC in the following, trained with either LS or WLSWGAN. Examples of generated amplitude spectral envelopes are shown in Fig.3. As baselines, we included the BLSTM+LS and the previous DCNN+WLSWGAN models. A separate listening test has been carried out for this assessment, with the same characteristics as the previous one in Sec. 4.1.

39 listeners took the test properly and Fig. 5 shows the aggregated results for all the voices (detailed results in Fig. 7.

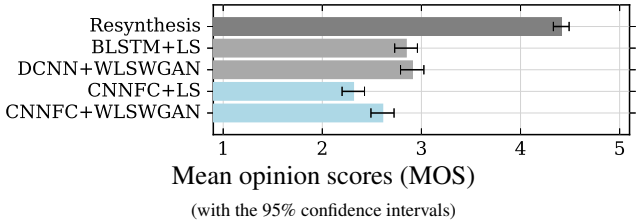

Fig. 5. MOS about perceived quality of a light model using: LS, or the suggested WLSWGAN optimization.

From Fig. 5, we can see that the CNNFC model does not obviously reach the same quality as the BLSTM and DCNN models. However, the quality is not so far away for a quite simpler architecture. Sound examples are also available online[4]. More importantly, the results do show that the WLSWGAN optimization improve the quality compared to the LS optimization (with p-value$<$0.01 on the means' difference). This is quite interesting since CNNFC can be trained with WLSWGAN in a few hours and provide a clearly better quality compared to the traditional LS training while imposing absolutely no modification at all at synthesis stage.

5. CONCLUSION

In this paper, we suggested a new loss for training a generator that is made of a spectrally weighted LS loss and a Wasserstein-trained discriminator's loss. It leads to a decent quality with only one hour of data and a computation time of approximately one day on a modern GPU. Compared to the original WGAN discriminator loss, we ensure a stable and consistent generation of the amplitude spectrum by emphasizing the importance of the LS loss in the low frequencies. Results of listening tests showed that the quality is indeed improved using the suggested loss. We can also note that we obtained these results without any cepstral compression, conversely to many vocoding approaches, nor deltas and MLPG algorithm. The exact same hyper-parameters were also used for the 10 voices used in the listening test. These simplifications are also important for freeing the researcher from unnecessary or cumbersome constraints.

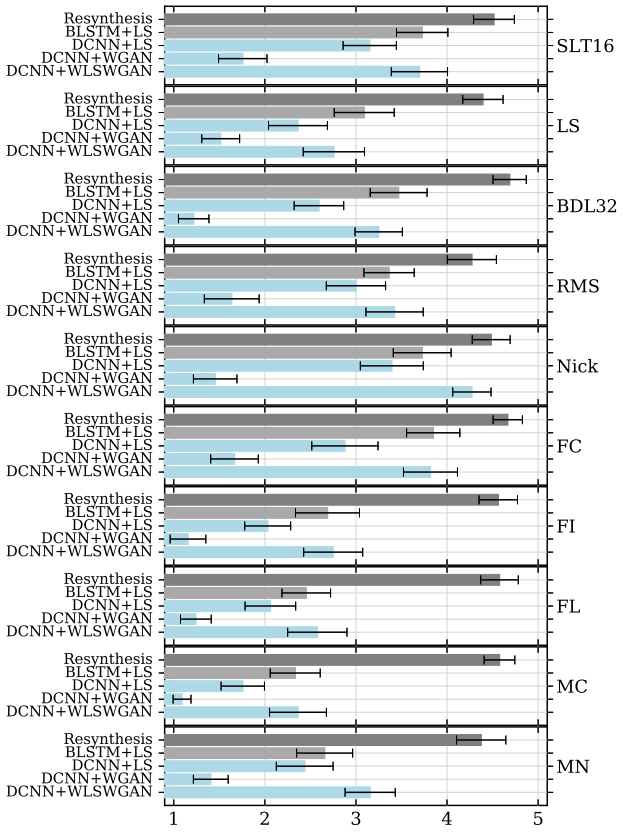

Fig. 6. MOS about the perceived quality of various models comparing: LS, WGAN and the suggested WLSWGAN.

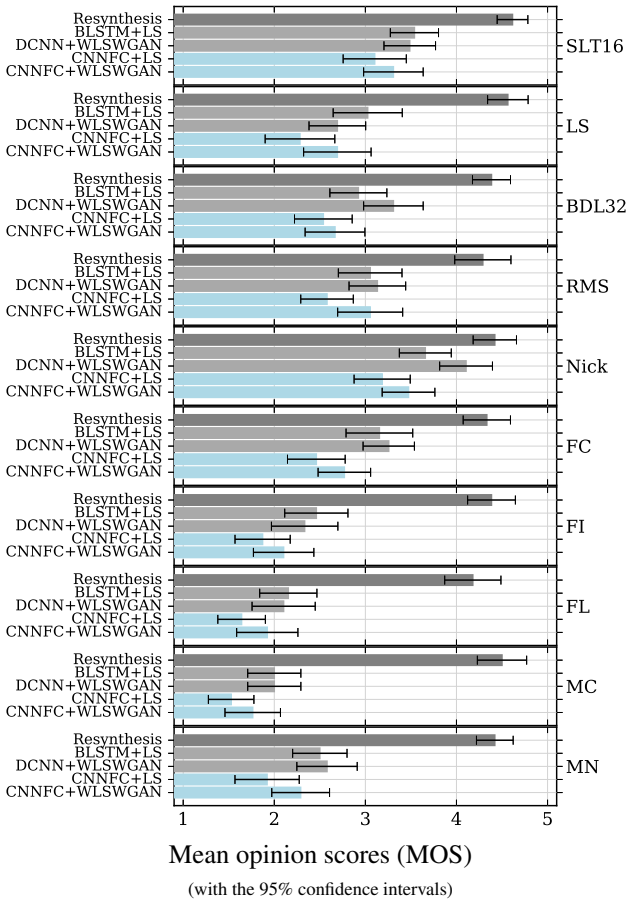

Fig. 7. MOS about the perceived quality of various models comparing: LS and the suggested WLSWGAN.

6. REFERENCES

[1] Heiga Zen, Andrew Senior, and Mike Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013, pp. 7962–7966.

[2] Simon King Zhizheng Wu, Oliver Watts, "Merlin: An open source neural network speech synthesis system," in *Proc. 9th Speech Synthesis Workshop (SSW9)*, 2016, pp. 218–223.

[3] H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "A hidden semi-Markov model-based speech synthesis system," *IEICE Trans. Inf. Syst.*, vol. E90-D, no. 5, pp. 825–834, 2007.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., pp. 2672–2680. Curran Associates, Inc., 2014.

[5] Martín Arjovsky, Soumith Chintala, and Léon Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learn-*

ing, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, 2017, pp. 214–223.

[6] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," *CoRR*, vol. abs/1609.03499, 2016.

[7] Y. Saito, S. Takamichi, and H. Saruwatari, "Statistical parametric speech synthesis incorporating generative adversarial networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 1, pp. 84–96, Jan 2018.

[8] Merlijn Blaauw and Jordi Bonada, "A neural parametric singing synthesizer," *CoRR*, vol. abs/1704.03809, 2017.

[9] G. Degottex, P. Lanchantin, and M. Gales, "A log domain pulse model for parametric speech synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 1, pp. 57–70, 2018.

[10] M. Airaksinen, L. Juvela, B. Bollepalli, J. Yamagishi, and P. Alku, "A comparison between straight, glottal, and sinusoidal vocoding in statistical parametric speech synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 9, pp. 1658–1670, Sept 2018.

[11] H. Kawahara, "Speech representation and transformation using adaptive interpolation of weighted spectrum: vocoder revisited," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1997, vol. 2, pp. 1303–1306.

[12] Yuchen Fan, Yao Qian, Feng-Long Xie, and Frank K. Soong, "TTS synthesis with bidirectional LSTM based recurrent neural networks.," in *Interspeech*, 2014, pp. 1964–1968.

[13] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A Vocoder-Based High-Quality Speech Synthesis System for Real-Time Applications," *IEICE Transactions on Information and Systems*, vol. 99, pp. 1877–1884, 2016.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, USA, 2012, NIPS'12, pp. 1097–1105, Curran Associates Inc.

[15] J. Kominek and A. W. Black, "The CMU ARCTIC speech databases," in *Proc. ISCA Speech Synthesis Workshop*, 2003, pp. 223–224, http://www.festvox.org/cmu_arctic.

[16] The Speech Synthesis Special Interest Group, "The Blizzard Challenge 2016 [Online]," `http://www.synsig.org/index.php/Blizzard_Challenge_2016/`, 2016.

[17] Martin Cooke, Catherine Mayo, and Cassia Valentini-botinhao, "Intelligibilityenhancing speech modifications: the Hurricane Challenge," in *in Proc. Interspeech*, 2013.

[18] D. Talkin, "REAPER: Robust Epoch And Pitch EstimatoR [Online]," by Google on Github: `https://github.com/google/REAPER`, 2015.

[19] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for hmm-based speech synthesis," in *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*, 2000, vol. 3, pp. 1315–1318.

[20] The ITU Radiocommunication Assembly, "ITU-R BS.1534: Method for the subjective assessment of intermediate quality levels of coding systems," Tech. Rep., ITU, 2003.

[21] The ITU Radiocommunication Assembly, "ITU-R BS.1284-1: En-general methods for the subjective assessment of sound quality," Tech. Rep., ITU, 2003.

[22] C. Callison-Burch and M. Dredze, "Creating speech and language data with amazons mechanical turk," in *Proc. of NAACL HLT Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk.* 2010, pp. 1–12, ACL.

[23] Maria K. Wolters, Karl B. Isaac, and Steve Renals, "Evaluating speech synthesis intelligibility using Amazon Mechanical Turk," in *Proc. 7th Speech Synthesis Workshop (SSW7)*, 2010, pp. 136–141.