Submission from CMU for Blizzard Challenge 2019

SaiKrishna Rallabandi, Peter Wu and Alan W Black

Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA.

{srallaba, peterw1, awb}@cs.cmu.edu

Abstract

In this paper we present the entry from CMU to Blizzard speech synthesis challenge 2019. We begin with a description of build process for our base voice. We then present the following modifications to base voice: (1) We investigate the effectiveness of sub-sentence training of acoustic models aimed at better utilization of available aligned data (2) We investigate the applicability of strategic gradient backpropagation to accelerate the training (3) We experiment with iterated dilated convolutions in WaveNet to obtain compact models. Although our current performance seems very inefficient, we are actively pursuing approaches to strengthen our voice building framework. We believe we are progressing in the right direction and anticipate a much stronger performance in the coming evaluations.

Index Terms: speech processing, convolutional neural networks, Tacotron, WaveNet

1. Introduction

Blizzard speech synthesis challenges were devised to better understand different corpus driven speech synthesis techniques on a common dataset. As a part of this, current evaluation focuses on building voices based on data resources from the internet. Language for the current evaluation is Mandarin. Our submission to this year's challenge was based on statistical parametric speech synthesis framework. Specifically, we have employed Sequence to Sequence neural network based approach to map the textual content to corresponding acoustics. There have been continuous and significant improvements in all aspects of this framework of speech synthesis from textual representations[1] through post filtering[2].

Approaches such as[3, 4, 5, 6] have demonstrated that Seq2Seq models are capable of reliably learning reasonable associations between the textual and acoustic modalities. These approaches have been utilized in building systems for new languages[7] as well as improving the models for existing ones[8]. Moreover, adaptation of these approaches to various tasks has been investigated[9]. Similarly, the world of vocoding has seen tremendous progress. Deep Neural Generative models aimed at vocoding[10] aim to model the joint probability of the data distribution and the conditioning information as a product of conditional distributions. Typical implementation of such models follows an autoregressive framework[10] although other formulations[11] have been suggested as well. Such models have proven very effective in addressing one of the major challenges with conventional vocoding techniques - fidelity. These advances have led to flexible systems capable of generating different styles[12, 13, 14, 13] of speech and ability to build voices from noisy[15] or very minimal data[16]. The elegance of such Seq2Seq models comes from the fact that they can be trained without making assumptions based on prior knowledge specific to speech. Therefore, we have employed Seq2Seq based approach as our cornerstone toward building our submission and have made extensions to the same.

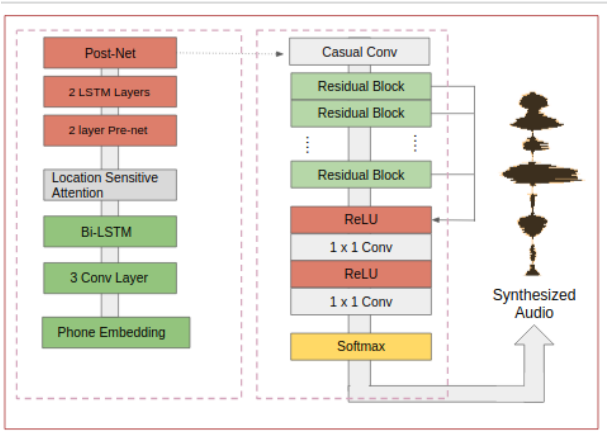

Figure 1: *Architecture of our BaseVoice build process*

Broadly, we have investigated the following approaches via our current submission:

- We investigate the possibility of jointly training acoustic model and the vocoder.
- Since the amount of aligned speech and text after our initial alignment step was limited, we investigate the applicability of sub sentence training in the context of acoustic models. We accomplish this using an external duration model at the phone level.
- To accelerate the training, we investigate the usage of strategic back propagation of gradients.
- To build compact and stable vocoders, we investigate building vocoders with shared parameters. Specifically, we share the parameters of all dilated convolutions with the same dilation rate in WaveNet.

The rest of this paper is organized as follows: We begin with a description of our base voice in section 2. We then present various approaches we investigated in section 3. This is followed by evaluation results and discussion. We present the observations made from post evaluation analysis and conclude this report.

2. BaseVoice

In this section, we describe our base voice built using a new module, FALCON Seq2Seq within Festvox[17] framework for statistical parametric speech synthesis. In brief, our system consists of two jointly trained components: (1) Acoustic model to predict acoustic vectors on a per frame basis and (2) Vocoder which generates speech on a per sample basis conditioned on the predicted acoustic vectors. For the current submission we

have not performed any postfiltering at the acoustic vector or sample level.

2.1. Data

The database used for building our submission was collected from an internet talk show by a well known Mandarin character. Topics of the utterances seemed contemporary and dealt with a variety of issues. We are provided 8 hours of speech data and corresponding transcription. Each utterance was one minute long, leading to a total of 480 utterances.

2.2. Voice Building

As preprocessing, we perform the following steps:

- *Alignment and Segmentation*: Since the provided utterances seemed too long to build a sequence to sequence acoustic model, we have performed segmentation of the utterances into smaller chunks(referred to as sub utterances hereafter) using EHMM based alignments. For acoustic models to perform alignment we have used multilingual models built as part of [18]. This resulted in 2178 sub utterances totalling three hours of speech. The mean duration of sub utterances was 5 seconds while the maximum utterance was of length 23 seconds. The normalized histogram of sub utterance durations can be seen in figure 2 (b).

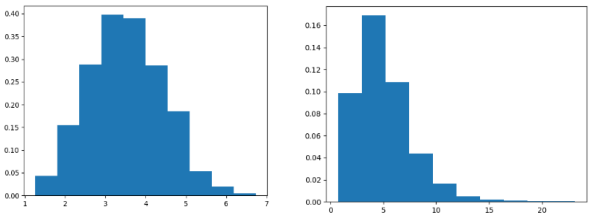

Figure 2: *(a) Normalized histogram plot of utterance lengths in Arctic dataset (b) Normalized histogram plot of sub utterance lengths from our alignment module*

- *Conversion of aligned text to Pinyin*: Our front end currently cannot process Mandarin characters. Hence we have converted the text data from the sub utterances into Pinyin.
- *Tokenization and G2P:* We consider any text entry separated by white space as a token. Once tokens are obtained, we have used US phoneset to perform mapping from graphemes to phonemes based on CMU pronunciation dictionary.
- *Acoustic Feature Extraction:* We perform acoustic feature vector extraction over a 50ms frames obtained by applying a hamming window with a frame shift of 12.5 msec. The speech files have been high pass filtered with cut off frequency of 60 Hz prior to feature extraction. We obtain 1025 dimensional linear and 80 dimensional mel frequencies.
- *Acoustic Model Component:* Our acoustic model is based on Tacotron[3] Seq2Seq speech synthesis system and is shown in the figure 1. We have used phones as the input instead of characters. We have not performed masking the loss value for padded frames as is typically done in Seq2Seq models. We found that not not masking forces to predict (zero) padded frames as well and helps the model better predict end of sentence as mentioned in[3]. Since adjacent frames seem to be correlated, our decoder predicts 3 frames per timestep. We have used a batch size of 64 to train the baseline model.
- *Vocoder:* Our vocoder is based on WaveNet[10]. Speech signal was power normalized and squashed to the range (-1,1). We have used 16 bit mulaw quantization to encode individual samples. Instead of transposed convolutions we have employed linear interpolation to upsample the acoustic frames to match the time resolution of speech samples. To optimize the model, we use discretized Mixture of Logistics loss[19, 20] with 12 logistic classes.

Our acoustic model and vocoder are trained jointly. While forcing the model to predict padded frames facilitates better prediction of end of sentence, authors are not aware of similar constraints to tighten the internal segment boundaries. We hypothesize that employing joint training would force local attention to better learn acoustic correspondences between segment boundaries. Our model consists of three loss components: (1) L1 Divergence between predicted and original mel spectrograms (2) L1 Divergence between predicted and original linear spectrograms (3) Mixture of Logistics loss from the vocoder. To make the training of vocoder faster, we have used chunks of randomly selected 8000 timesteps of raw signal and corresponding predicted acoustic frames. We have used a batch size of 16 and an upsample convolution block with 4 layers that upsample at $\{2,2,4$ and $5\}$ times respectively.

3. Experiments

3.1. Sub sentence Training using aligned segment durations

From figure 2, it can be observed that sub utterances from our alignment module are skewed towards longer lengths. In our informal model component evaluations, we observed that the acoustic model training seemed unstable compared to other models we built using English data. Inspection of attention mechanism led us to believe that the model was failing to converge. We hypothesized that this might be due to the wide variety in the length of our sub utterances. Based on analysis in section 3.1.1, we postulate that a reasonable way to circumvent this issue is to use sub-sentence training: To train smaller chunks within the sub-utterances since we already have duration information from our alignment module.

3.1.1. Role of Attention in Sub Utterance Alignment

Let ph_1,...,ph_m denote the phonemes in the textual domain that have been transformed by an encoder network to state vectors $p_1, ..., p_m$. Let y_1,...,y_n denote acoustic frames in the target sequence. A typical attention based encoder decoder network such as Tacotron factorizes the joint probability of acoustic frames $Pr(y_1, ..., y_n|p_1...p_m)$ as $\Pi_{t=1}^{t=n}$ Pr($y_t \mid p_1$...m,s_t) where s_t is a decoder state summarizing y_1,...y_{t1}. For each time step t, an attention variable a_t is used to denote which encoded phoneme state of $p_1...p_m$ aligns with y_t. Let P(a_t = j|$p_1...p_m$,s_t) denote the probability that encoder state p_j is relevant for output y_t. Typically this conditional probability is estimated using a softmax function over attention scores computed from p_j and decoder state s_t as follows.

$$P(a_t = j|p_1...p_m, s_t) = \frac{e^{A_\theta(p_j, s_t)}}{\sum_{k=1}^{t} e^{A_\theta(p_j, s_t)}} \quad (1)$$

where A_θ is the attention unit that scores each input state p_j as per the decoder state s_t. This is followed by a convex combination of the input states to model log likelihood for each output acoustic vector y_t.

$$logPr(y_t|x_1..x_m) = logPr(y_t|\sum_a P_t(a)x_a) \quad (2)$$

In this scenario, attention is essentially a latent deterministic variable that is conditionally dependent on the convex combination from encoded representation of input phonemes. It is responsible for the sentence internal association between textual and acoustic modalities.

3.1.2. Sub Sentence Training

We posit that it is possible to improve the association learnt by acoustic model by using sub sentence training: selecting aligned segments of text and acoustics within a sentence. In addition, selecting segments within a sentence might lead to the model utilizing the available data more efficiently. We note that such an approach is already used for vocoding: Typical vocoders the authors are aware of are trained using aligned chunks of acoustic vectors and corresponding speech samples as opposed to full utterances. While this is due to GPU memory constraints in the context of vocoders, we investigate applying similar strategy to acoustic modeling. We belive that this facilitates local attention mechanism to better learn the mapping from phonetic space to the acoustic space.

The steps we have used for sub sentence training are mentioned below:

Algorithm 1 Sub sentence Training of Acoustic Model

```
Obtain segment durations using EHMM
Refine segment durations using MoveLabel
while n < GlobalUpdates do
  for instance in batch do
    StartIdx = random(0, SubUtteranceLength)
    StartSegment,StartDur=findClosest(StartIdx,AlignedSegments)
    EndIdx = StartIdx + SelectedDuration
    EndSegment,EndDur=findClosest(EndIdx,AlignedSegments)
  end for
  Add selected sub sentence <text,frames> pair to batch
  Iterate over batch
end while
```

3.1.3. Refinement of Segment Boundaries

Since alignment between text and acoustic vectors is crucial for sub sentence training, we refine the segment durations initially obtained using EHMM using [21].

- To improve the segment boundaries obtained from initial labeling, we employ different speech representation, MCEPs. For each of the states obtained from segmentation, we extract acoustic feature vectors over a 5ms frames obtained by applying a hamming window. Spectral representation that we use is MCEPs and were extracted using the SPTK toolkit [22]. The order of MCEP was chosen to be 24 with a frequency warping factor of 0.42 and a small value (1.0E-08) was added to the periodogram. For F0, we interpolate between unvoiced section ensuring breaks during silences and then apply a post smoothing using a 25 ms window.
- We examine each segment boundary and consider moving it forward or backward (by one frame) and investigate whether this decreases the distance between original and predicted frame. This process is performed over all the labels and then the models are rebuilt. The distance is measured in terms of unnormalized Mel Cepstral Distortion(MCD) including the energy coefficient but not the deltas. We have performed 10 iterations over the entire database as the improvement in MCD stopped at that point. The results of this procedure have been outlined in the table 1.

Table 1: *Refinement of segment durations*

Pass	No. of Moves	MCD	F0 Error	Duration Error
1	58941	7.065	32.808	0.964
2	55467	7.049	32.712	0.964
3	51224	7.047	32.7	0.963
4	48930	7.041	32.4	0.959
5	47844	7.036	32.35	0.948
6	47342	7.022	32.23	0.946
7	46541	7.021	31.91	0.942
8	43451	7.018	31.82	0.942
9	42872	7.020	31.86	0.947

3.2. Strategic Backpropagation to enable joint training

While training our base model, inspection of scalar loss value indicated that the model reaches around 95% of the optimal loss value within 10K steps(5% of training steps). Subsequent gradient updates result in minute contributions towards the transition toward optimal loss value. The authors have not found any existing literature analyzing this region in the context of Seq2Seq Text to speech models. We hypothesize that the model compensates for macro issues(eg., pronunciation of vowels vs consonants) in the initial stages and spends rest of training phase in filling the micro details (eg., pronunciation of *ax* vs *ah*).

We posit that contribution from smaller gradient updates in the *micro detail* phase can be ignored to result in better training: Since the absolute magnitude of the gradients is small while they contribute equal weight to normalization, ignoring such gradients might lead to sharper overall update helping the model in the *micro detail* phase. Note that L1 divergence which is employed in Seq2Seq TTS models also results in sharper gradients compared to L2 divergence which was employed in segment based statistical parametric methods.

To facilitate this, we select the gradient updates to include in the model training based on the absolute loss values after each forward pass during training. The steps are mentioned in Algorithm 2. This approach is inspired by selective back propagation[23]. However, they disable backpropagation using a hard rule: When the threshold hyperparameter is set to 0, the gradient is backpropagated only when the model makes an error. While this may be suitable in the context of classification tasks for which their model was deployed, tasks involving regression

Algorithm 2 Strategic Back-propagation

```
Forward pass through the model
Compute divergence values for each instance in batch.
loss=[]
Pick a threshold value
for divergence in BatchOfDivergences do
  if divergence > threshold then
    loss+=divergence
  else
    idx=random(0,1)
    if idx > 0.5 then
      loss+=[divergence]
    end if
  end if
end for
BatchLoss=sum(loss)/len(loss)
Back propagate loss
```

present an opportunity to investigate interesting modifications. Our approach can be seen as a softer version, similar to [24]: When absolute value of a gradient is less than preset threshold, its inclusion in the backpropagation is stochastically determined. Other variants of this approach can be employed such as scheduling the threshold as well as the stochasticity. However, in our current submission we have employed a simple stochasticity component: If a gradient is less than the threshold, it is included in the backpropagation based on a coin toss.

Our approach can also be interpreted from the view of selective attention[25] to the most relevant contributors[26]. In its extreme case, choosing to only use the gradient with the most loss magnitude is similar computationally to hard attention. Such an approach has been shown to be employed in natural neural network design (WTA approach in [27]).

3.3. Dilated Convolution Parameter Sharing in WaveNet

WaveNet [10] is an autoregressive neural model with a stack of 1D convolutional layers that is capable of directly generating audio signal. The input to WaveNet is subjected to corresponding gated activations while passing through each dilated convolutional layer and is classified by the final softmax layer into a μ law encoding. The concrete form of the residual gated activation function is given by following equation:

$$r_d(x) = tanh(W_f * x) \odot \sigma(W_g * x) \tag{3}$$

where x and $r_d(x)$ are the input and output with dilation d, respectively. The symbol $*$ is a convolution operator with dilation d and the symbol $\odot$ is an element-wise product operator. W represents a convolution weight. The subscripts f and g represent a filter and a gate, respectively. The joint probability of a waveform $\mathbf{X}$ can be written as:

$$P(X|\theta) = \prod_{t=1}^{T} P(x_t|x_1, x_2..x_{t-1}, \theta) \tag{4}$$

given model parameters θ. During implementation of WaveNet, the autoregressive process is realized by a stack of dilated convolutions. The final output y_t at time step t can be expressed mathematically as:

$$\hat{y_t} \sim \sum_{d=0}^{D} h_d * r_d(x) \tag{5}$$

where x, y represent input and output vectors; D is the number of different dilation used and d is the dilation factor; h_d is the convolution weights. This stack of convolutions is repeated multiple times in the original WaveNet. Optimization in WaveNet is performed based on the error between predicted sample and the ground truth sample conditioned on previous samples in the receptive field alongside the local conditioning. Expressing the loss function being optimized mathematically the error at sample t is:

$$l_t = Div(\hat{y_t}||y_t) \tag{6}$$

Here, we define the divergence similar to the [19], To optimize this loss, the contribution from the individual convolution layers towards this global error function must be nullified. Now let us consider the expression for intermediate output for a single filter in Eqn 5:

$$x_{out}(t) = \sum_{\tau=0}^{t} h(\tau)x(t-\tau) \tag{7}$$

where τ is the receptive field covered by the model and $h(\tau)$ represents the discrete state representation at time t. Without loss of generality and dropping the term τ for brevity, the spectral representation generated by the model can be expressed as:

$$Y(z) = H(z)X(z) \tag{8}$$

Considering the discrete nature of input from Eqn 6, an interpretation of Eqn 8 is that the neural auto regressive model acts as the transfer function and is discretized by convolving with the samples from original signal. It has to be noted that this is similar to the formulation of source filter model of speech, specifically the periodic components corresponding to voiced sounds. Voiced sounds typically represented as impulse train are convolved with the transfer function to generate spectral envelope. As a corollary, from equation 6 and 8, we posit that the optimization in WaveNet model is performed by minimizing the divergence between true and approximate spectral envelopes. With the presented interpretation, we hypothesize that sharing parameters across the filters(individual dilated convolution components) facilitates the filters to be more stable compared to a non shared scenario. This is also inspired by observations that show efficiency of such parameter sharing [28, 29]. In terms of model size, we obtain a reduction of two thirds due to parameter tying.

4. Evaluation

The subjective evaluation was conducted based on various categories: pleasantness, speech pauses, stress, intonation, emotion, listening effort. The identifier of our system is P. Mean opinion score of our system as provided by all listeners is depicted in figure 3.

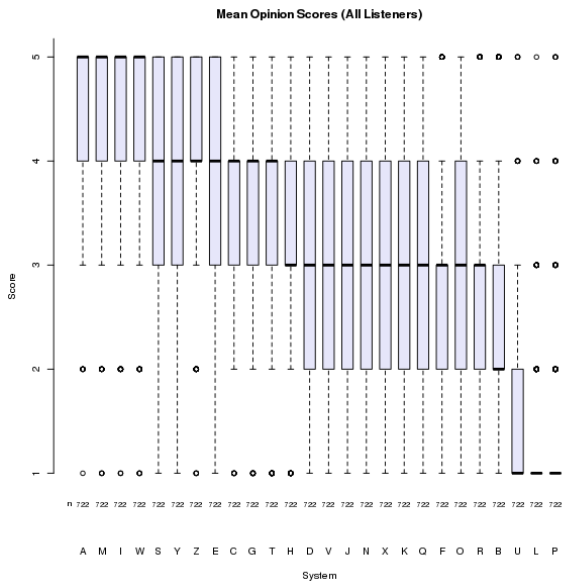

Figure 3: *MOS Scores for all listeners - Overall Impression*

4.1. Discussion of Results

We have ranked last in the current evaluation. Informal evaluations of our submission revealed that speech generated by our model lacks comprehensibility. To analyze the reason for this, we have performed an internal evaluation of individual approaches proposed in the current submission. For this we have used Arctic[30], LJSpeech[31] and Indic[32] datasets. Although these datasets differ in the language of the current evaluation and the findings might not be transferable, we have chosen them for initial analysis since we have worked with these datasets in the past. Based on these ablation evaluations, we currently believe that the component responsible for most degradation of quality in our system is that of sub sentence training.

4.1.1. Sub Sentence Training - Only Acoustic Model

We performed sub sentence training of acoustic model using Arctic and Indic datasets. The duration of sub sentences was varied from 1.5 to 8 seconds. We have observed that while during training the model seems to learn association between the textual input and acoustic vectors, there were instances where the synthesized speech was either distorted or incoherent with the text. Most of these instances were found when the ratio of selected duration to the mean length in the dataset was less than 40%. In addition, a histogram analysis showed that the sub utterances obtained from our alignment module are significantly longer compared to the other datasets we have employed attention based models thus far. For comparision, a histogram plot of Arctic dataset can be seen in figure 2. While the length distribution was not as regular as Arctic for the other dataset(LJSpeech) we compared against, we observed that the maximum length in [31] was still less than in our sub utterances.

Here are some of the things we believe we should improve for our next submission:

- Data size from the output of our alignment module was 3 hours compared to 8 hours that was originally provided. Although we implemented sub sentence approach to handle this, we believe a much stronger approach would be to investigate approaches to use the whole data by realignment.
- Our current implementation of jointly training acoustic model and vocoder seems very naive: (1) Data resource utilisation by vocoder in such a training paradigm is very inefficient. (2)Vocoder always receives predicted spectrogram. We believe (a)scheduling the vocoder input by starting from original spectrogram but slowly transitioning to predicted spectrogram (b) adopting wake sleep training procedure as opposed to the current implementation might lead to a better base architecture.
- We plan to further investigate the performance of sub sentence based training approach for acoustic model.

5. Conclusion

In this paper we have presented the entry from CMU to Blizzard speech synthesis challenge 2019. We have made these modifications to our previous submission:(1) We investigate the effectiveness of sub-sentence training of acoustic models aimed at better utilization of available data (2) We investigate the applicability of selective gradient backpropagation to accelerate the training (3) We experiment with iterated dilated convolutions in WaveNet to obtain compact models. We are actively pursuing approaches to strengthen our voice building framework. We believe we will have a much stronger framework and hence a more competitive submission in the coming evaluations.

6. References

[1] K. Kastner, F. Santos, Y. Bengio, and A. Courville, "Representation mixing for tts synthesis," 2019.

[2] Y.-C. Wu, K. Kobayashi, T. Hayashi, P. L. Tobing, and T. Toda, "Collapsed speech segment detection and suppression for wavenet vocoder," *arXiv preprint arXiv:1804.11055*, 2018.

[3] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech synthesis," *arXiv preprint arXiv:1703.10135*, 2017.

[4] A. Gibiansky, S. Arik, G. Diamos, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, "Deep voice 2: Multi-speaker neural text-to-speech," in *Advances in neural information processing systems*, 2017, pp. 2962–2970.

[5] W. Ping, K. Peng, and J. Chen, "Clarinet: Parallel wave generation in end-to-end text-to-speech," *arXiv preprint arXiv:1807.07281*, 2018.

[6] F. Biadsy, R. J. Weiss, P. J. Moreno, D. Kanvesky, and Y. Jia, "Parrotron: An end-to-end speech-to-speech conversion model and its applications to hearing-impaired speech and speech separation," *arXiv preprint arXiv:1904.04169*, 2019.

[7] Y. Choi, Y. Jung, Y. Kim, Y. Suh, H. Kim *et al.*, "An end-to-end synthesis method for korean text-to-speech systems," *Phonetics and Speech Sciences*, vol. 10, no. 1, pp. 39–48, 2018.

[8] Y. Lee and T. Kim, "Robust and fine-grained prosody control of end-to-end speech synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5911–5915.

[9] B. Bollepalli, L. Juvela, and P. Alku, "Speaking style adaptation in text-to-speech synthesis using sequence-to-sequence models with attention," *arXiv preprint arXiv:1810.12051*, 2018.

[10] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[11] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," *arXiv preprint arXiv:1811.00002*, 2018.

[12] Y. Wang, R. Skerry-Ryan, Y. Xiao, D. Stanton, J. Shor, E. Battenberg, R. Clark, and R. A. Saurous, "Uncovering latent style factors for expressive speech synthesis," *arXiv preprint arXiv:1711.00520*, 2017.

[13] R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. J. Weiss, R. Clark, and R. A. Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," *arXiv preprint arXiv:1803.09047*, 2018.

[14] Y. Wang *et al.*, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," *arXiv preprint arXiv:1803.09017*, 2018.

[15] W.-N. Hsu, Y. Zhang, R. J. Weiss, Y.-A. Chung, Y. Wang, Y. Wu, and J. Glass, "Disentangling correlated speaker and noise for speech synthesis via data augmentation and adversarial factorization," 2018.

[16] Y. Chen *et al.*, "Sample efficient adaptive text-to-speech," *arXiv preprint*, 2018.

[17] A. W. Black, "Clustergen: A statistical parametric synthesizer using trajectory modeling," in *Ninth International Conference on Spoken Language Processing*, 2006.

[18] A. Black, "Cmu Wilderness Multilingual Speech Dataset," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019.

[19] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma, "Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications," *arXiv preprint arXiv:1701.05517*, 2017.

[20] L. Juvela, B. Bollepalli, J. Yamagishi, and P. Alku, "Waveform generation for text-to-speech synthesis using pitch-synchronous multi-scale generative adversarial networks," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6915–6919.

[21] A. W. Black and J. Kominek, "Optimizing segment label boundaries for statistical speech synthesis," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 3785–3788.

[22] S. W. Group *et al.*, "Speech signal processing toolkit (sptk)," *http://sp-tk. sourceforge. net*, 2009.

[23] S. Bendelac, "Enhanced neural network training using selective backpropagation and forward propagation," Ph.D. dissertation, Virginia Tech, 2018.

[24] A. Jiang *et al.*, "Accelerating deep learning by focusing on the biggest losers," *arXiv preprint*, 2019.

[25] B. Graham, "Fractional max-pooling," *arXiv preprint arXiv:1412.6071*, 2014.

[26] S. Shankar, S. Garg, and S. Sarawagi, "Surprisingly easy hard-attention for sequence to sequence learning," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 640–645.

[27] S. Dasgupta, C. F. Stevens, and S. Navlakha, "A neural algorithm for a fundamental computing problem," *Science*, vol. 358, no. 6364, pp. 793–796, 2017.

[28] E. Strubell *et al.*, "Fast and accurate entity recognition with iterated dilated convolutions," 2017.

[29] H. Inan, K. Khosravi, and R. Socher, "Tying word vectors and word classifiers: A loss framework for language modeling," *arXiv preprint arXiv:1611.01462*, 2016.

[30] J. Kominek and A. W. Black, "The cmu arctic speech databases," in *Fifth ISCA workshop on speech synthesis*, 2004.

[31] K. Ito *et al.*, "The lj speech dataset," 2017.

[32] A. Baby, "Resources for indian languages," in *CBBLR workshop, International Conference on Text, Speech and Dialogue,2016*.