A TensorFlow Implementation of DC-TTS: yet another text-to-speech model

I implement yet another text-to-speech model, dc-tts, introduced in Efficiently Trainable Text-to-Speech System Based on Deep Convolutional Networks with Guided Attention. My goal, however, is not just replicating the paper. Rather, I'd like to gain insights about various sound projects.

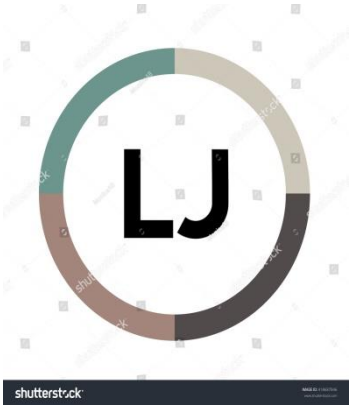
Github: https://github.com/RamsteinWR/dt_tts

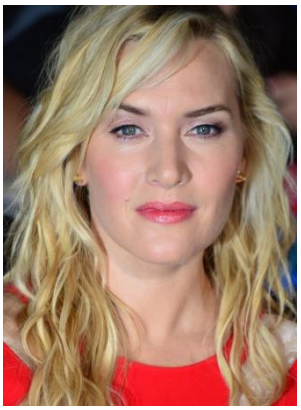
Requirements

- NumPy >= 1.11.1
- TensorFlow >= 1.3 (Note that the API of `tf.contrib.layers.layer_norm` has changed since 1.3)
- librosa
- tqdm
- matplotlib
- scipy

Data

train English models and an Korean model on four different speech datasets.





1. LJ_Speech Dataset 2. Nick_Offerman's Audiobooks 3. Kate_Winslet's_Audiobook 4. KSS_Dataset

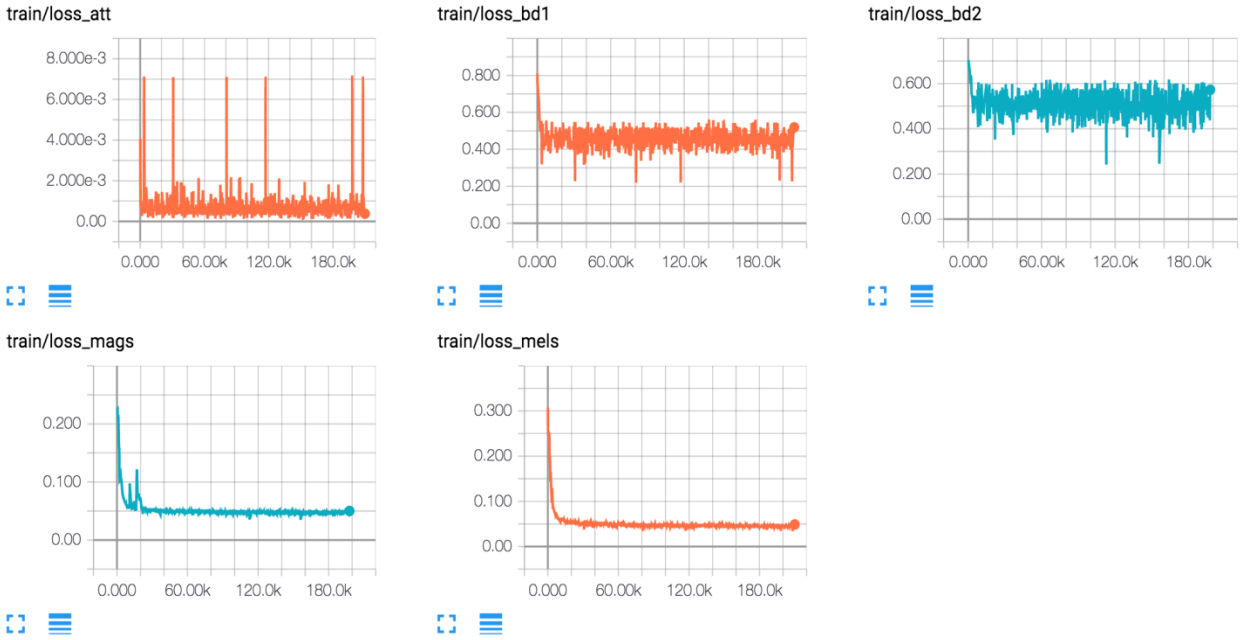
LJ Speech Dataset is recently widely used as a benchmark dataset in the TTS task because it is publicly available, and it has 24 hours of reasonable quality samples. Nick's and Kate's audiobooks are additionally used to see if the model can learn even with less data, variable speech samples. They are 18 hours and 5 hours long, respectively. Finally, KSS Dataset is a Korean single speaker speech dataset that lasts more than 12 hours.

Training

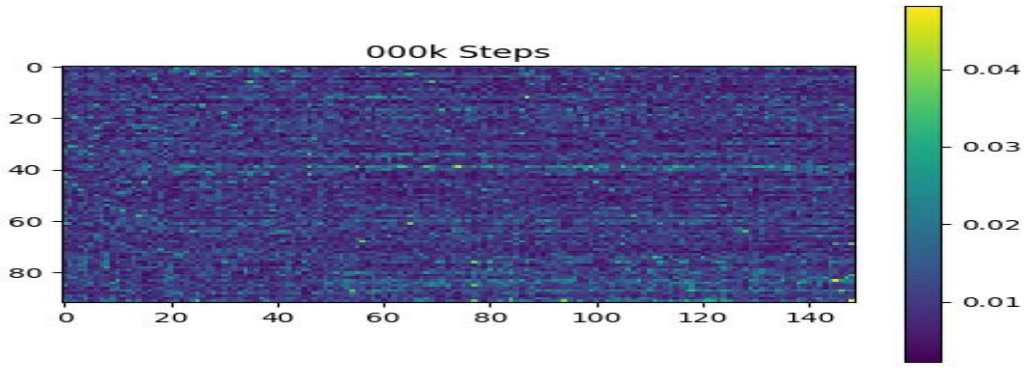
- **S**TEP 0. Download LJ Speech Dataset or prepare your own data.
- STEP 1. Adjust hyper parameters in `hyperparams.py`. (If you want to do preprocessing, set prepro True`.
- STEP 2. Run `python train.py 1` for training Text2Mel. (If you set prepro True, run python prepro.py first)
- STEP 3. Run `python train.py 2` for training SSRN.

You can do STEP 2 and 3 at the same time, if you have more than one gpu card.

Training Curves



Attention Plot



Sample Synthesis

I generate speech samples based on Harvard Sentences as the original paper does. It is already included in the repo.

- Run `synthesize.py` and check the files in `samples`.

Generated Samples

Dataset	Samples
LJ	50k 200k 310k 800k
Nick	40k 170k 300k 800k
Kate	40k 160k 300k 800k
KSS	400k

Pretrained Model for LJ

Download this.

Notes

- The paper didn't mention normalization, but without normalization I couldn't get it to work. So I added layer normalization.
- The paper fixed the learning rate to 0.001, but it didn't work for me. So I decayed it.
- I tried to train Text2Mel and SSRN simultaneously, but it didn't work. I guess separating those two networks mitigates the burden of training.
- The authors claimed that the model can be trained within a day, but unfortunately the luck was not mine. However obviously this is much fater than Tacotron as it uses only convolution layers.
- Thanks to the guided attention, the attention plot looks monotonic almost from the beginning. I guess this seems to hold the aligment tight so it won't lose track.
- The paper didn't mention dropouts. I applied them as I believe it helps for regularization.
- Check also other TTS models such as Tacotron and Deep Voice 3.