The Emotional Speech Generation Final Report

Yao Fu, yf2470

1. Introduction

This semester we tried multiple approaches for the Emotional Speech Generation/ Speech Emotion Transfer task. We first investigates the features statistics of different emotions. Based on the assumption that the Speech Emotion Transfer is comparable with the Voice Conversion task (one is to convert the speaker identity, the other is to transfer the emotion), we start our project by following the VC models. We start from simple Feed Forward Networks and Convolutional Neural Networks, then implement the more advanced Transformer model, and demonstrate its effectiveness of modeling voice. However, the transformer model with an emotion embedding alone cannot achieve emotion transfer, so we try to equip it with Variational Inference and Adversarial Training. After multiple training attempts and hyper-parameter tuning, we find out they just not work. So we switch to another larger Chinese dataset, and test the above models again. However, the model still does not work. Then we take a step back and see how the VC model is trained, we find out that the VC model can work with simple speaker embedding approach. Then we perform the cross-verification: we train the VC model on the emotion dataset, and we train our emotion model on the VC dataset. We find out that the VC model, previously works on the VC dataset, does not work on the emotion dataset, while our emotions model, previously cannot work on the emotion dataset, now can work on the VC dataset even at its minimal setting (simply use the speaker embedding, no Variational Inference, no Adversarial Training). We reach the conclusion that, the emotion transfer task is indeed more challenging than the VC task. One need to consider larger datasets and different paradigms for this task.

In the following sections, we first outline the timeline for this project. Then we discuss the challenges and the conclusions of our attempts. Finally we discuss the future directions.

2. The Project Timeline

2.1 Feature Analysis

We start our project by examine the features of 16 different emotions on the EPSaT dataset. We can see different feature distribution for different emotions, but none of they may strong enough to dominate one emotion. We try to use the preliminary features to train a classifier (f0, speech rate, intensity .etc), but the result is not as good. Later Brenda directed me a whole feature set for emotion classification of which the total number of features is over 1K. The feature analysis simply gives a direct taste of speech since I am somehow new to this area. Then we dive directly to the modeling.

2.2 Preliminary Models

In the modeling process, we assume that a speech can be decomposed into a content portion and an emotion portion. We want to disentangle the content and the emotion into vector

representations, as is shown in the VC task and other machine learning tasks. After the disentanglement, we want to change the emotions vector while maintaining the content vector. A straightforward method is, during training, we first encode the speech into a vector, then decoder this vector with its own emotion embedding. During test time, we change the emotion embedding to another emotion's. This is essentially an Autoencoder. So we first train the Autoencoders to verify if a neural network is able to model the speech spectrum.

Our experiments show that simple feed forward neural networks can perform frame-wise autoencoding. However, when we change the emotion embedding during test time, the model cannot transfer the emotion. To further test the more advanced models, we implemented a CNN model and a Transformer model. We find out that the Transformer model is the best in terms of Autoencoding. But still, it cannot perform conversion with the change of emotion vector.

Here we list the models we use at this stage:

- The Feed Forward Network
- The Convolutional Neural Network
- The Transformer Model

2.3 Variational Autoencoders and Generative Adversarial Networks

Since we cannot perform transfer simply by changing the emotion vector, following the VC approach, we try two more advanced techniques: a). the Variational Autoencoders and b). the Generative Adversarial Networks. However, even we tried multiple hyper-parameters and training techniques, we cannot get a convergence. Here we list the models we tried:

- The Transformer model with Variational Autoencoding:
    - Use a Gaussian Prior, try multiple KL-annealing parameters.
- The Transformer model Adversarial Training
    - Try multiple discriminator and discriminator combinations, including a single Real/ Fake discriminator, an emotion classifier, and the combination of the two.
    - Try multiple hyper-parameters, including: different coefficients for the reconstruction loss, the discriminator loss, and the generator loss.
    - Try multiple training techniques, including: with/ without pretrain, different training times of the discriminator and the generator.

2.4 Switch to Another Dataset

After the above attempts, we suspect that the problem may come from the dataset. So we change to another Chinese dataset and perform the above steps again. However, the models still do not work. To find out what is going on, we switch to the VC task as we assume the emotional transfer may be comparable with VC from the very beginning.

2.5 Comparison of Voice Conversion

We train a VC model and find out everything works fine. Then we do a cross-examination: we try to use the VC model on the emotional generation dataset, and try to train our models on the VC dataset. We find out that when trained on our emotional dataset, the VC model cannot work.

But when we train our model on the VC dataset, our model works fine. This result indicate that the two tasks may not comparable. The VC task is indeed easier to learn. In our experiments, we find out only use the simplest emotion embedding approach, the model can work. We do not need the advanced Variational Autoencoding and the Adversarial Training for the VC task. However, even with the advanced techniques, the model still cannot work on the emotion transfer task.

3. Discussions

When things do not work in deep learning, one may think of three things: a). the dataset b). the capability of the model itself c). the optimization/ training method. Here we discuss them shortly.

3.1 The Modeling Capability

In our Autoencoding task, we find out the Transformer model is the most powerful model among the three with the smallest L1 loss. In fact, it also achieves remarkable effectiveness on other tasks (Machine Translation, Summarization). So it might be the best model we have at hand. One may also try to improve it by designing models specially for speech (like WaveNet). But intuitively I do not think this is where the problem is.

3.3 The Difficulty of Training GANs

It is difficult to train GANs. First, one may not able to find an effective discriminator since the signal from the discriminator is too sparse. Second, even if the discriminator can classify different emotions, the generator may trick it by generating noise. This is also shown in the Image domain as the Adversarial Attacks. Third, it is difficult to monitor the training process. The loss is not an indicator of the model convergence because the generator and the discriminator consistently push the loss towards their own directions. All the above factors make GANs difficult to train, even with multiple attempts to help training.

4. Future Directions

4.1 Larger Datasets

The above discussions show that we may not be able to improve from neither the model side nor the optimization side. So I think that a larger dataset with more explicit emotion might be a direction worth to try.

4.2 The TTS Approach

Another approach is to first recognize the input speech into text, then perform emotional text speech. The Amazon Alexa team have shown that the emotional TTS is possible even with simple emotion embedding. However, this approach may also requires a large dataset.

5. Summary

In this project, we explored multiple model architectures for the emotional speech generation task. We point out that the model capacity may not be a bottleneck for this task. One may need to find better datasets and more advanced GAN training techniques.