TOWARDS A CUSTOMIZABLE TEXT-TO-SPEECH PERSONAL ASSISTANT

Chenyu Shi
s3500063

Siwen Tu
s3631400

Shuang Fan
s3505847

Shupei Li
s3430863

ABSTRACT

The audio assistant technology has been widely applied based on the high development of the deep learning field. In this project, we use deep learning models to accomplish a customizable text-to-speech personal voice assistant. Its main features are twofold — multi-language support and multi-speaker support. We trained the neural networks on audio datasets and fine-tuned them in different languages with multi-speaker settings. The final model can produce realistic audio based on input texts, and users can customize the type of languages or speaker identities according to their personal preferences.

***K*eywords** VITS · TTS · Deep Learning

1 Introduction

With the rapid development of the deep learning field, text-to-speech techniques are widely applied in our daily lives. For example, they are used in audiobook reading, voice guidance in navigation systems, and human-machine interfaces. One of the most important applications is the voice assistant, such as Siri on the iPhone or Cortana on Windows. These voice assistants can provide help and advice in a speech format, which is more convenient and acceptable than text for humans.

Many languages are spoken worldwide, and people prefer voice assistants to use a familiar language. Additionally, individuals have varying preferences for the type of voice the assistants use, favoring human-like voices over a cold mechanical sound. Hence, customizing voice assistants is a promising direction for this technology.

In this project, we utilize the VITS model [1] to design and implement a customizable text-to-speech personal voice assistant. VITS is a deep learning model for the text-to-speech task based on a conditional variational autoencoder. Our main contributions are as follows. Firstly, we train the VITS model from scratch in Chinese to achieve the multi-language objective since the original VITS implementation was designed for English. In addition, we fine-tune the model based on pre-trained weights in different languages, with a primary focus on Mandarin and Japanese. It further improves the voice assistant's capability — better output resolution and more language support. Moreover, we fine-tune the model on various types of speakers to complete the goal of multi-speaker. We added five new speakers for Chinese and six for Japanese. The final result is a personal voice assistant that allows users to select their preferred languages and voice types. Through training and evaluation in the experiment, our final model can produce realistic audio based on the input text data, serving as an effective customizable text-to-speech voice assistant.

The rest of the reports will be presented in three sections. In Section 2, we will show and explain the methodology used in the models. In Section 3, the experiment results will be displayed and discussed. And in Section 4, we will have a conclusion of the report and project.

2 Methodology

We describe VITS [1], the backbone of our TTS assistant, from the perspective of variational inference in detail. We also introduce VITS's architecture and training process to provide a comprehensive overview.

2.1 CVAEs

The variational autoencoder (VAE), proposed in [2], is a popular approach widely used in unsupervised learning. It is an effective function approximator and can be optimized by the standard stochastic gradient descent (SGD) method. It has shown promising application value in various fields, such as complex pattern recognition, segmentation, future prediction from static images, etc [3].

Conditional variational autoencoders (CVAEs) are variants of VAEs. Remember that VAEs learn posterior distribution parameters from dataset x without any label information. In CVAEs, we have additional contextual information c when estimating the posterior distribution. The objective function of the CVAE can be written as follows:

$$\max \quad \log p_\theta\left(x|c\right) - \mathcal{D}\left[q_\phi\left(z|x\right) \| p_\theta\left(z|x\right)\right] \tag{1}$$

where $p_\theta(x|c)$ denotes the marginal log-likelihood of the data, $p_\theta(z|x)$ is the posterior distribution and $q_\phi(z|x)$ is its approximator. According to the definition of Kullback-Leibler divergence and Bayes' theorem, maximizing Equation 1 is equivalent to maximizing the following equation:

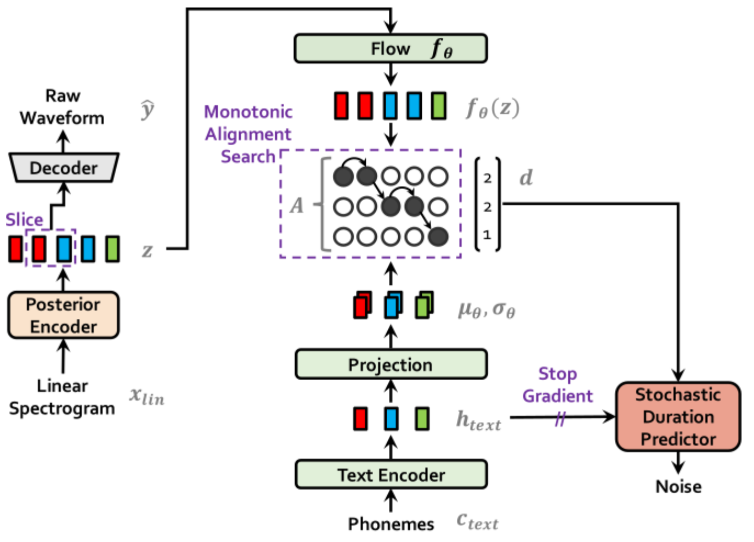
$$\max \quad \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x \mid z) - \log \frac{q_\phi(z \mid x)}{p_\theta(z \mid c)}\right] \tag{2}$$

Notice that the relative KL divergence is always a non-negative number. We can derive the evidence lower bound (ELBO) of the objective function based on Equation 1 and Equation 2:

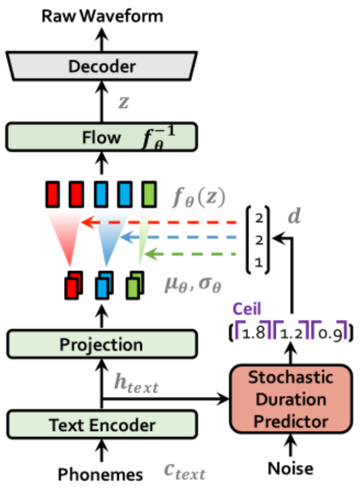
$$\begin{aligned} \log p_\theta(x \mid c) &\geq \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x \mid z) - \log \frac{q_\phi(z \mid x)}{p_\theta(z \mid c)}\right] \\ &= \underbrace{\mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta\left(x|z\right)\right]}_{\text{Reconstruction loss}} - \underbrace{\mathcal{D}\left[q_\phi(z \mid x) \| p_\theta(z \mid c)\right]}_{\text{Regularization loss}} \end{aligned} \tag{3}$$

The ELBO comprises two parts — the reconstruction loss and the regularization loss. The reconstruction loss describes the data distribution given the latent space, while the regularization loss measures the divergence between the true prior distribution and the encoder's approximate distribution. In practice, we usually choose to minimize the negative ELBO because the term $\mathcal{D}\left[q_\phi\left(z|x\right) \| p_\theta\left(z|x\right)\right]$ is hard to calculate. However, a reasonable posterior approximator $q_\phi(z|x)$ can effectively alleviate the impact of the bias.

2.2 Expressing VITS as a CVAE



(a) Training procedure



(b) Inference procedure

Figure 1: The architecture of VITS. This figure is directly adapted from [1].

The architecture of VITS is shown in Figure 1. VITS model can be expressed as a CVAE. We will illustrate the main idea behind VITS based on the reconstruction loss as well as the regularization loss.

Reconstruction Loss. During the training, we have ground truth soundtracks x and their corresponding text c_{text}. Kim et al. [1] define the reconstruction loss as the L_1 norm between the input's mel-spectrogram and the estimated mel-spectrogram produced by the model:

$$L_{recon} = \|x_{mel} - \hat{x}_{mel}\|_1 \tag{4}$$

It is easy to compute the mel-spectrogram x_{mel} of a specific audio file. As for the approximate mel-spectrogram $\hat{x}_{mel}$, we can upsample the latent space to the waveform domain and then transform it to the mel-spectrogram domain.

Regularization Loss. The text-to-speech task requires an automatic alignment between the text input and the corresponding voice features. VITS adopts the monotonic alignment search algorithm to align the audio and textual embeddings. The monotonic alignment search algorithm (MAS) is proposed in [4]. The intuition behind the MAS is applying the dynamic programming algorithm to maximize the likelihood of the data distribution. It will produce an alignment matrix A. The condition c in VITS is defined as the concatenation of the text input c_{text} and the alignment matrix A, i.e., $c = [c_{text}, A]$. Given the condition c, we can calculate the regularization loss following the definition of KL-divergence:

$$\begin{gathered} L_{kl} = \mathcal{D}\left[q_\phi(z \mid x) \| p_\theta(z \mid c)\right] = \log q_\phi\left(z \mid x_{lin}\right) - \log p_\theta\left(z \mid c_{\text{text}}, A\right) \\ z \sim q_\phi\left(z \mid x_{lin}\right) = N\left(z; \mu_\phi\left(x_{lin}\right), \sigma_\phi\left(x_{lin}\right)\right) \end{gathered} \tag{5}$$

It is worth mentioning that VITS uses the linear-scale spectrogram of the soundtrack x_{lin} instead of the mel-spectrogram. [1] explains that the linear-scale spectrogram is helpful in enhancing the resolution of generated audio. To improve the model's performance further, VITS introduces a normalizing flow f_θ that connects the conditional prior distribution and the latent space in the following way:

$$p_\theta(z \mid c) = N\left(f_\theta(z); \mu_\theta(c), \sigma_\theta(c)\right)\left|\det \frac{\partial f_\theta(z)}{\partial z}\right| \tag{6}$$

Equation 6 transforms the latent space into some complex distributions that possess a more powerful expressiveness, which is critical for generating realistic outputs.

2.3 Training VITS

The major part of VITS is a CVAE. However, there are more things to consider when handling the text-to-speech task. Firstly, we need to decide the duration of each token in outputs. A straightforward idea is adding a deterministic duration predictor into the model. But this naive method has a big flaw — it can not reflect the features of different speakers, which limits the model's application in multi-speaker scenarios. Alternatively, VITS designs a stochastic duration preditor whose loss function is L_{dur} and optimizes it with other modules during training.

VITS also integrates adversarial training into the architecture to guide the optimization direction. With a discriminator D and a decoder G, VITS combines two kinds of loss in the adversarial training, namely the least-squares loss and the additional feature-matching loss:

$$\begin{aligned} L_{adv}(D) &= \mathbb{E}_{(y,z)}\left[(D(y)-1)^2 + (D(G(z)))^2\right] \\ L_{adv}(G) &= \mathbb{E}_z\left[(D(G(z))-1)^2\right] \\ L_{fm}(G) &= \mathbb{E}_{(y,z)}\left[\sum_{l=1}^{T} \frac{1}{N_l}\left\|D^l(y) - D^l(G(z))\right\|_1\right] \end{aligned} \tag{7}$$

Taken together, we can obtain the final loss for VITS training:

$$L_{vae} = L_{recon} + L_{kl} + L_{dur} + L_{adv}(G) + L_{fm}(G) \tag{8}$$

During the inference stage, we fix all weights and disgard the posterior encoder because we only have the text input. Final outputs are produced by the decoder G.

3 Experiments

The hardware and software we use in the experiment are listed below. Our official project website is `https://sd12321sd.github.io/api_project.github.io/`. The code of our project is available on `https://github.com/ShupeiLi/api-final-project/tree/master`.

Hardware:

- 1 Titan Xp GPU

Software:

- Cython==0.29.21
- librosa==0.8.0
- matplotlib==3.3.1
- numpy==1.18.5
- phonemizer==2.2.1
- scipy==1.5.2
- tensorboard==2.3.0
- torch==1.6.0
- torchvision==0.7.0
- Unidecode==1.1.1
- chardet
- pypinyin
- monotonic_align
- protobuf==3.20.0

3.1 Training from Scratch

To validate the effectiveness of the VITS model, we train the model on the Chinese Standard Mandarin Speech Corpus and record the performance of different training phases.

Preprocessing Due to the distinctive structure of Chinese characters and their relationship with pronunciation, which differs significantly from English, we need to preprocess the raw data containing audio and Chinese text to better align with the training process. We convert Chinese text into representations using the International Phonetic Alphabet (IPA). We establish mappings for Latin to Bopomofo symbols (a phonetic script used in Mandarin Chinese transliteration and writing systems), Bopomofo symbols to Romaji, and Bopomofo symbols to IPA. This involves leveraging external libraries such as *pypinyin*, *jieba*, and *cn2an* to convert Arabic numerals, Latin characters, and Chinese characters into Bopomofo symbols. Additionally, various regular expressions are employed to locate and replace specific elements in the Chinese text. These procedures aim to optimize the data for the training process, facilitating the establishment of a model that captures the relationship between audio and Chinese text.

Experiment Setting We train the model for around 96,000 steps, lasting about 48 hours. We record the performance of 0 step, 5,000 steps, 10,000 steps, 30,000 steps, 50,000 steps, 70,000 steps and 96,000 steps to demonstrate a remarkable improvement, which is shown on our project website.

Dataset Chinese Standard Mandarin Speech[1]. This dataset consists of 10,000 sentences which are spoken by a Chinese female who pronounces standard Mandarin. The average length of the sentences is around 16 characters and the audio is recorded at 48kHz 16-bit sampling frequency, in PCM WAV format. The transcribed text of the audio is recorded as a txt file.

3.2 Fine-tune the Model

To build a versatile model capable of handling multiple languages and speakers, we perform fine-tuning on the model using separate datasets for Chinese and Japanese speakers. The customized text-to-speech personal assistant, developed through this process, is showcased on our project website.

Preprocessing A bit different from the preprocessing of the Chinese dataset, we also built a pipeline for converting the Japanese text to IPA. We build mappings for Romaji to IPA, consonant to their representations. We utilize the *pyopenjtalk* library to convert the Japanese text to Romaji with the accent, and then we deal with some special consonants and convert the Romaji to IPA.

Experiment Setting Based on the model trained in Section 3.1, we fine-tune the model for around 84,500 steps, which lasts for about 16 hours. We provide five different speakers including child, middle-aged female, middle-aged male, young female and young male speakers for Chinese. Additionally, we include six speakers for the Japanese language, spanning child, young female, young male, middle-aged female, middle-aged male, and aged male voices. The outcomes of this training process are showcased on our project website.

Chinese Dataset: AISHELL-3 for Chinese [5]. This dataset is a large-scale multi-speaker Mandarin speech corpus provided by Beijing Shell Shell Technology Co., Ltd. The corpus contains roughly 88,000 utterances which are spoken by 218 native Chinese Mandarin speakers. These speakers have different genders, ages, and accents which are marked in the corpus.

[1]https://www.data-baker.com/open_source.html

Japanese Dataset: JVS corpus [6]. This dataset consists of Japanese transcripts and multi-speaker audio data. These audio files are recorded by 100 professional speakers who are of different gender and age. The audio is sampled at 24 kHz and the average duration is 10 seconds. For each speaker, they provide 100 different recordings.

4 Conclusion

To conclude, we have built a customizable text-to-speech personal audio assistant based on the VITS model and fine-tuned it to achieve optimal performance. Our final model can produce realistic audio based on the content of the input text. Additionally, users can select their preferred languages and types of voices. The model finally works with satisfying performance and efficiency as a personal voice assistant.

For our model, there are several aspects that could potentially be improved in the future. First of all, the number of languages and types of voices in our model is currently limited. However, increasing the supported languages and speakers does not pose significant challenges for the model, considering the existence of the large open-source text-to-speech community. We just need to repeat each step in our experiment with a new audio dataset of a new language or speaker to extend our model. Additionally, the model does not perform equally well on all types of voices and languages in our experiment. To address this issue, we could find and use higher-quality audio datasets for training and fine-tuning the model with increased computing resources.

References

[1] Jaehyeon Kim, Jungil Kong, and Juhee Son. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5530–5540. PMLR, 18–24 Jul 2021.

[2] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022.

[3] Carl Doersch. Tutorial on variational autoencoders, 2021.

[4] Jaehyeon Kim, Sungwon Kim, Jungil Kong, and Sungroh Yoon. Glow-tts: A generative flow for text-to-speech via monotonic alignment search. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 8067–8077. Curran Associates, Inc., 2020.

[5] Xin Xu Shaoji Zhang Ming Li Yao Shi, Hui Bu. Aishell-3: A multi-speaker mandarin tts corpus and the baselines. 2015.

[6] Shinnosuke Takamichi, Kentaro Mitsui, Yuki Saito, Tomoki Koriyama, Naoko Tanji, and Hiroshi Saruwatari. Jvs corpus: Free japanese multi-speaker voice corpus. *arXiv preprint arXiv:1908.06248*, Aug 2019.