1J-04

VITS を用いた TTS 音声合成の性能評価

青山 柊惟† 片桐 孝洋‡ 大島 聡史‡ 永井 亨‡
名古屋大学 情報学部 コンピュータ科学科† 名古屋大学 情報基盤センター‡

1. はじめに

近年，TTS(Text-to-Speech)音声合成の発展がめざましい．その中でも VITS(Variational Inference with adversarial learning for end-to-end Text-to-Speech)[1] はそれ以前のシステムよりも優れた音質を持つことで知られている．

本研究では，VITS についての性能評価を，GPU スーパーコンピュータ環境で行うことを目的とする．

2. VITS

VITS はエンドツーエンド並列 TTS 音声合成システムである．VITS 登場前の TTS システムは二段階 TTS システムが有力であった．ここで，二段階とは，テキストを中間表現に変換するシステムと中間表現から音声を出力するシステムの 2 種類に分かれたシステムを指す．テキストから直接音声を出力するエンドツーエンドのシステムは精度が二段階システムに劣っていた． VITS は一段階目と二段階目を特徴変数でつなぐことによってエンドツーエンドシステムでありながら二段階システムを凌駕する性能を持つに至った．

VITS は推論速度についての他モデルとの比較は行われたが，複数 GPU についての学習速度の評価は行われてこなかった．そこで本研究では,VITS の学習速度について，GPU スパコンを用いて評価することを目的にする．特に GPU の数を変化させて性能評価を行う．

3. 評価方法

3.1 実験データ

本実験では，LJSpeech データセット[2]を用いて学習の性能評価を行った．LJSpeech とは，7 冊のノンフィクション本を一人の英語話者によって読み上げた 13100 本の短い音声データからなる．

VITS のソースコードは Github 上で公開されている公式による実装[3]を用いた．

Performance Evaluation of Text-to-Speech Synthesis using VITS
† Shui Aoyama, Department of Computer Science, School of Informatics Nagoya University
‡Takahiro Katagiri, Satoshi Ohshima, Toru Nagai, Information Technology Center, Nagoya University

3.2 実験環境

本実験では，名古屋大学情報基盤センターに設置されているスーパーコンピュータ「不老」TypeⅡサブシステムを用いた．詳細は表 1 の通りである．

表 1. 「不老」TypeⅡサブシステムの 1 ノード性能の詳細

CPU	Intel Xeon Gold 6230, 20 コア, 2.10 - 3.90 GHz × 2 ソケット
GPU	NVIDIA Tesla V100 (Volta) SXM2, 2,560 FP64 コア, upto 1,530 MHz × 4 ソケット(4GPU)
メモリ	メインメモリ(DDR4 2933 MHz) 384 GiB (32 GiB × 6 枚 × 2 ソケット), デバイスメモリ(HBM2) 32 GiB × 4 ソケット(4GPU)
理論演算性能	倍精度 33.888 TFLOPS (CPU 1.344 TFLOPS × 2 ソケット, GPU 7.8 TFLOPS × 4 ソケット)
GPU 間接続	NVLINK2 (1GPU から他の 3GPU に対してそれぞれ 50GB/s×双方向)

VITS は 2021 年 6 月 14 日に更新されたバージョンを利用した． python3.9.12, CUDA11.4 を用いた．Python ライブラリは GitHub 上の requirements.txt に従い導入したが，protobuf のみ 3.20.1 を導入した．

VITS における機械学習のハイパパラメタは，GitHub 上の指示に従い ljs_base.json を用いて設定した．また，バッチサイズは 64 に設定した．

3.3 テストプログラムの概要

本実験では，train.py を用いて学習速度を評価した．このプログラムはモデルファイルを出力する．ここで生成されたモデルファイルを用いることで TTS 音声合成を行うことができる．

実行する際のバッチジョブスクリプトにおいて環境変数である CUDA_VISIBLE_DEVICES を"0","0,1"等と変更することによって GPU 数を変化させた.

```
def main():
    (中略)
  mp.spawn(run, nprocs=n_gpus, args=(n_gpus, hps,))
  for epoch in range(epoch_str, hps.train.epochs + 1):
    timecounter = time.time()
    if rank==0:
      train_and_evaluate( (略) )
    else:
      train_and_evaluate( (略) )
    print(time.time() - timecounter)
    scheduler_g.step()
    scheduler_d.step()
```

上は train.py の一部である. 灰色部にコードを追加し, epoch 一つあたりの学習時間の計測を可能とした.

4. 実験結果

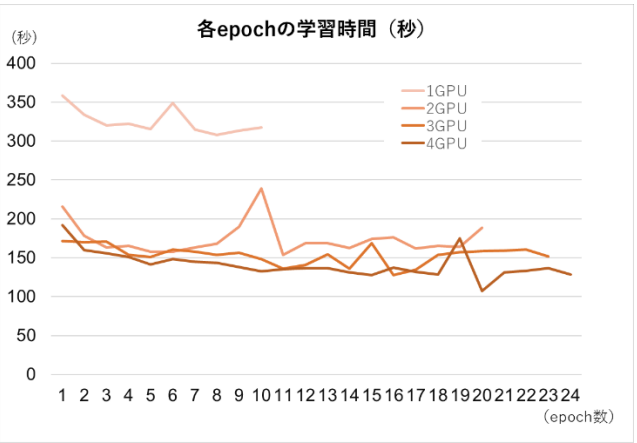

図 1　GPU 数あたりの 各 Epoch の学習時間[秒]
(実行時間 1 時間で動作する最大のエポック数)

図 1 より, 1GPU と 2GPU の間では明確な速度差が確認でき, 最大で 2.2 倍ほど高速化されている. 一方, 2GPU～4GPU では大きな差が見られない. なぜこのような差が生じるかの分析は, 今後の課題である.

一方, GPU 数を増やすと, 1 時間で実行できるエポック数は, ごくわずかであるが増加している. これは, GPU を使うほど高速化している理由からであり, 機械学習の観点からスケールするといえるため, 複数 GPU 利用の効果があるといえる.

5. おわりに

本発表では, VITS における TTS 音声合成の学習速度を, 最新の GPU スーパーコンピュータである「不老」Type Ⅱ サブシステムで評価した. その結果, 1GPU から 2GPU までの実行時間加速の効果はあるが, 複数 GPU における学習速度は大きく変わらないことが明らかとなった. この理由の解析等の詳細は, 発表時に紹介する予定である.

一方, 本稿で実行した機械学習のハイパパラメタはデフォルト実行であり, 学習率の向上の観点から, チューニングの余地がある. このハイパパラメタチューニングは, ソフトウェア自動チューニング技術[4]適用の重要な課題の一つである. ハイパパラメタチューニングにおける学習結果の影響への調査は, 重要な今後の課題である.

謝辞

本研究は JSPS 科研費 JP19H05662 の助成を受けたものです. また, 研究内容についてコメントを頂いた星野哲也准教授に感謝の意を表します.

参考文献

[1] Kim, J., Kong, J., and Son, J. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. arXiv preprint arXiv:2106.06103, 2021.

[2] https://keithito.com/LJ-Speech-Dataset/ (閲覧日:2023 年 1 月 11 日)

[3] https://github.com/jaywalnut310/vits (閲覧日:2023 年 1 月 10 日)

[4] T. Katagiri, D. Takahashi, Japanese Autotuning Research: Autotuning Languages and FFT, Proc. of the IEEE, Vol. 106, Issue 11, pp. 2056-2067 (2018)