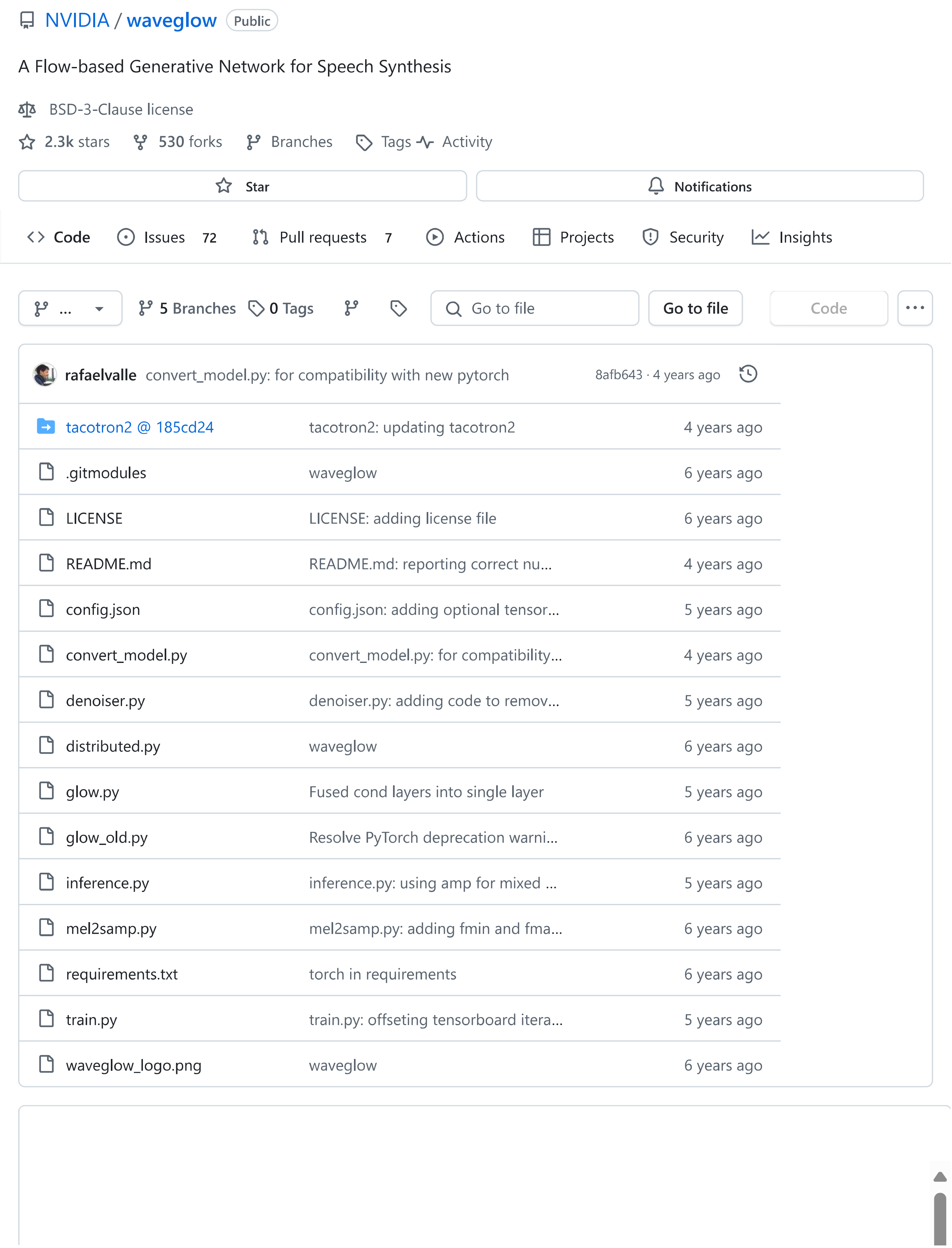

NVIDIA / waveglow

A Flow-based Generative Network for Speech Synthesis

BSD-3-Clause license

2.3k stars · 530 forks · Branches · Tags · Activity

rafaelvalle convert_model.py: for compatibility with new pytorch · 8afb643 · 4 years ago

tacotron2 @ 185cd24	tacotron2: updating tacotron2	4 years ago
.gitmodules	waveglow	6 years ago
LICENSE	LICENSE: adding license file	6 years ago
README.md	README.md: reporting correct nu...	4 years ago
config.json	config.json: adding optional tensor...	5 years ago
convert_model.py	convert_model.py: for compatibility...	4 years ago
denoiser.py	denoiser.py: adding code to remov...	5 years ago
distributed.py	waveglow	6 years ago
glow.py	Fused cond layers into single layer	5 years ago
glow_old.py	Resolve PyTorch deprecation warni...	6 years ago
inference.py	inference.py: using amp for mixed ...	5 years ago
mel2samp.py	mel2samp.py: adding fmin and fma...	6 years ago
requirements.txt	torch in requirements	6 years ago
train.py	train.py: offseting tensorboard itera...	5 years ago
waveglow_logo.png	waveglow	6 years ago

WaveGlow: a Flow-based Generative Network for Speech Synthesis

Ryan Prenger, Rafael Valle, and Bryan Catanzaro

In our recent paper, we propose WaveGlow: a flow-based network capable of generating high quality speech from mel-spectrograms. WaveGlow combines insights from Glow and WaveNet in order to provide fast, efficient and high-quality audio synthesis, without the need for auto-regression. WaveGlow is implemented using only a single network, trained using only a single cost function: maximizing the likelihood of the training data, which makes the training procedure simple and stable.

Our PyTorch implementation produces audio samples at a rate of 1200 kHz on an NVIDIA V100 GPU. Mean Opinion Scores show that it delivers audio quality as good as the best publicly available WaveNet implementation.

Visit our website for audio samples.

Setup

1. Clone our repo and initialize submodule

   ```
   git clone https://github.com/NVIDIA/waveglow.git
   cd waveglow
   git submodule init
   git submodule update
   ```

2. Install requirements `pip3 install -r requirements.txt`

3. Install Apex

Generate audio with our pre-existing model

1. Download our published model
2. Download mel-spectrograms
3. Generate audio `python3 inference.py -f <(ls mel_spectrograms/*.pt) -w waveglow_256channels.pt -o . --is_fp16 -s 0.6`

N.b. use `convert_model.py` to convert your older models to the current model with fused residual and skip connections.

Train your own model

README BSD-3-Clause license

2. Make a list of the file names to use for training/testing

```
ls data/*.wav | tail -n+10 > train_files.txt
ls data/*.wav | head -n10 > test_files.txt
```

3. Train your WaveGlow networks

```
mkdir checkpoints
python train.py -c config.json
```

   For multi-GPU training replace `train.py` with `distributed.py` . Only tested with single node and NCCL.

   For mixed precision training set `"fp16_run": true` on `config.json` .

4. Make test set mel-spectrograms

```
python mel2samp.py -f test_files.txt -o . -c config.json
```

5. Do inference with your network

```
ls *.pt > mel_files.txt
```

Releases

No releases published

Packages

No packages published

Contributors 7

Languages

- ● Python 100.0%