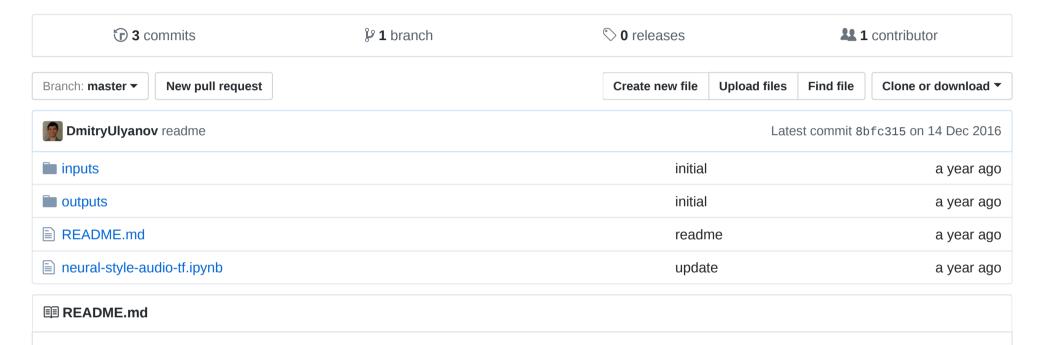
DmitryUlyanov / neural-style-audio-tf

TensorFlow implementation for audio neural style.

#tensorflow #style-transfer #neural-style



Audio Style Transfer

This is a TensorFlow reimplementation of Vadim's Lasagne code for style transfer algorithm for audio, which uses convolutions with random weights to represent audio features.

To listen to examples go to the blog post. Also check out Torch implementation.

So far it is CPU only, but if you are proficient in TensorFlow it should be easy to switch. Actually it runs fast on CPU.

Dependencies

- python (tested with 2.7)
- TensorFlow (installation instructions)
- librosa

```
pip install librosa
```

numpy and matplotlib

The easiest way to install python is to use Anaconda.

How to run

- Open neural-style-audio-tf.ipynb in Jupyter.
- In case you want to use your own audio files as inputs, first cut them to 10s length with:

```
ffmpeg -i yourfile.mp3 -ss 00:00:00 -t 10 yourfile_10s.mp3
```

- Set content_filename and style_filename in the third cell of Jupyter notebook to your input files.
- Run all cells.

The most frequent problem is domination of either content or style in the output. To fight this problem, adjust ALPHA parameter. Larger ALPHA means more content in the output, and ALPHA=0 means no content, which reduces stylization to texture generation. Example output outputs/imperial_usa.wav, the result of mixing content of imperial march from star wars with style of U.S. National Anthem, was obtained with default value ALPHA=1e-2.

References

- Original paper on style transfer: A Neural Algorithm of Artistic Style
- Neural style TensorFlow implementation
- Publications on texture generation with random convolutions:
- Extreme Style Machines
- Texture Synthesis Using Shallow Convolutional Networks with Random Filters
- A Powerful Generative Model Using Random Weights for the Deep Image Representation