Muzip: Music Compression Using Neural Networks

Michael Arruza-Cruz, David Morales, Divya Saini

Background

Music files are traditionally relatively large and difficult to process, so compression methods are often used to **reduce the file size**. MP3 files are one such compressed representation that results in lost data.

Our project seeks to use convolutional neural networks (CNNs) to **convert audio files into a compressed representation** that reduces file size, while still allowing r**econstruction of the original file**. Unlike prevailing literature that tends to center around spectogram representations of an audio file, we attempt to investigate whether it is possible to garner higher quality and better compressed output on **raw audio signals**.

The aim of this music compression is to reduce the redundancy in a song in order to be able to **store, transmit, and search** for the song at **low bit rates.**

Dataset

We use the **FMA music analysis dataset**, which provides 917 GiB of audio from 106,574 tracks from 16,341 artists and 14,854 albums of 161 genres. Along with this audio, the dataset provides pre-computed features with track- and user-level metadata, tags, and free-form text.

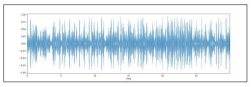

Figure 1. Audio sample from the FMA dataset.

For rapid iteration, we use the smaller version of the dataset containing 8,000 30-second snippets taken from a **multitude of songs across 8 balanced genres, in MP3 format**.

Approach

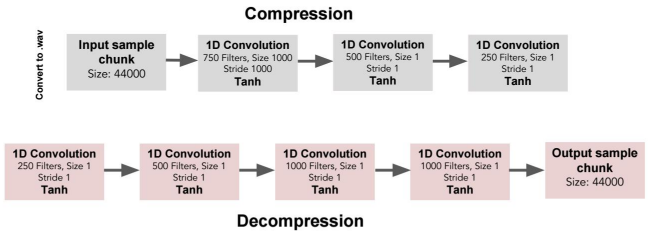

Figure 2. Model Design. We use a convolutional neural network using multiple one-dimensional convolutions over the inputs. **Grey boxes show compression** layers and **pink show decompression** layers.

Experiments & Results

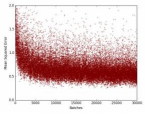

Figure 3. MSE over batches during training.

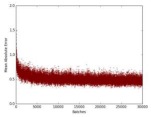

Figure 4. MAE over batches during training.

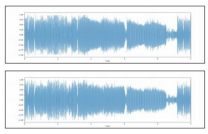

Figure 5. Example 5-second output. **Top:** Actual sample. **Bottom:** Decompressed sample

Model	MSE	MAE	R^2
3 Compression Layers 4 Decompression Layers	0.0040	0.0388	0.8561
4 Compression Layers 5 Decompression Layers	0.0010	0.0182	0.9639

Table 1. Model performance. Performance metrics of our two best music compression and decompression models on **131 examples**. Both provide **4x compression**.

The first model evaluated was designed as described in the approach section above. The second model we evaluated was designed similarly but slightly deeper. The architecture is described below.

Encoder:

```
(Conv1D(150, 1, strides = 1, activation='tanh'))
(Conv1D(100, 1, strides = 1, activation='tanh'))
(Conv1D(50, 1, strides = 1, activation='tanh'))
```

Decoder:

```
(Conv1D(50, 1, strides = 1, activation='tanh'))
(Conv1D(100, 1, strides = 1, activation='tanh'))
(Conv1D(150, 1, strides = 1, activation='tanh'))
(Conv1D(200, 1, strides = 1, activation='tanh'))
(Conv1D(200, 1, strides = 1, activation='linear'))
```

Using a standard laptop computer and the **Keras** framework, compression and decompression of a **30-second sample takes 64 milliseconds**.

Features

We **convert each MP3 track to WAV** format, which yields a **vector of floats between -1 and 1** representing the audio. Because the vectors representing each 30-second track is quite large (1,320,000 values), we **split the WAV file into 30, 1-second chunks**. Each of these chunks is used as a training example. To improve training, signal values are multiplied by 10.

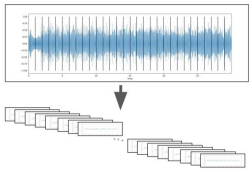

Figure 6. Audio sample is split into shorter chunks.

Conclusion & Future Work

- Convolutional Neural Networks are able to **compress music samples to a fourth of original size.**
- Decompressed samples are **noisy, but preserve most of the underlying music** sample.
- **Taking genre into account** may result in better decompression (network will learn to account for similarities in genre).
- **Evaluating semantic information captured** in compressed representations may reveal potential use for compressed samples.

References

[1] https://github.com/mdeff/fma

[2] https://github.com/michaelArruza/CS230-Compression

[3] Michael Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. Fma: A Dataset for Music Analysis, 2016.

[4] Feng Jiang, Wen Tao, Shaohui Liu, Jie Ren, Xun Guo, and Debin Zhao. An End-to-End Compression Framework Based on Convolutional Neural Networks, 2017.

[5] Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo. Learning Multi-Attention Convolutional Neural Network for Fine-Grained Image Recognition.