Generative Audio Samples

Model Card

# Overview

The model is a Variational Auto-Encoder trained on a data set of 950 audio-samples of acoustic and synthesized musical instruments, limited to the note C4.

This is effectively two models:

* An encoder: which takes a sample and translates it to a latent space
* A decoder: which takes a vector in the latent space and generates samples from it.

Whilst the model can to some extent regenerate samples in the training dataset, the main goal is to generate randomised samples. There are many ways to do this, interpolating between known samples, randomising around a known sample, or completely random.

The model could also be used to encode a user-provided sample, and then generate new samples around that.

# Inputs & Outputs

## Encoder

Input: an audio sample of a single note, playing middle-C, at 44.1 kHz using the standard .wav format.

Output: a small vector of floats in the approximate range [-3, 3]

Note that the model can process multiple samples in a single batch.

## Decoder

Input: a small vector of numbers, each from a distribution with mean 0 and standard deviation 1.

Output: 2 seconds of audio at 44.1 kHz in the standard .wav format.

## Formatting

In practice the encode & decoder need a wrapper around them as the internal representation used is a windowed short-term Fourier transform (STFT).

# Model Architecture

Several architectures have been evaluated, but they conform to the following:

* A Recursive Neural Network (RNN) applied at each time-step of the STFT to generate a set of control parameters (encoder), or generate an STFT (decoder) from the control parameters.
* A Variational Auto-Encoder: mapping the sequence of control parameters to a small number of variables in a latent space.

# Limitations

This model is strictly applicable to single samples of a musical instrument at the frequency of middle-C, ie: 261.63 Hz.

Performance will degrade if applied to tones whose spectrum deviates too much from this frequency and its harmonics.

The model may also perform poorly on samples with large random variations in timbre over time, for example synthesized tones with high resonance moving arbitrarily.

# Performance

The model achieves a loss of 94.6 when encoding and decoding the samples in the test data-set which it has not seen before.

This number is tricky to interpret. It means that the mean square error over a normalised STFT of 1024 x 57 time buckets is 94.6, ie: 0.16%

However we’re dealing with audio, the same signal at 10 or 100 times lower amplitude will be perceived to have the same content, just less loud.

**The current perceptual performance is way below what should be achievable with more work.**