Abstract

Methodology for designing and training neural network autoencoders for applications involving musical audio is proposed. Two topologies are presented: an autoencoding sound effect that transforms the spectral properties of an input signal (named ANNe); and an autoencoding synthesizer that generates audio based on activations of the autoencoder's latent space (named CANNe). In each case the autoencoder is trained to compress and reconstruct magnitude short-time Fourier transform frames. When an autoencoder is trained in such a manner it constructs a latent space that contains higher-order representations of musical audio that a musician can manipulate.

With ANNe, a seven layer deep autoencoder is trained on a corpus of improvisations on a MicroKORG synthesizer. The musician selects an input sound to be transformed. The spectrogram of this input sound is mapped to the autoencoder's latent space, where the musician can alter it with multiplicative gain constants. The newly transformed latent representation is passed to the decoder, and an inverse Short-Time Fourier Transform is taken using the original signal's phase response to produce audio.

With CANNe, a seventeen layer deep autoencoder is trained on a corpus of C Major scales played on a MicroKORG synthesizer. The autoencoder produces a spectrogram by activating its smallest hidden layer, and a phase response is calculated using phase gradient heap integration. Taking an inverse short-time Fourier transform produces the audio signal.

Both algorithms are lightweight compared to current state-of-the-art audio-producing machine learning algorithms. Metrics related to the autoencoders' performance are measured using various corpora of audio recorded from a MicroKORG synthesizer. Python implementations of both autoencoders are presented.