ANNs in Feedback Paths for novel Audio Effects

Patrik Lechner

IC\M/T
FH Stp.

June 19 2020

https://github.com/hrtlacek/audioFeedbackNN

Background

The presented system was built around the following ideas:

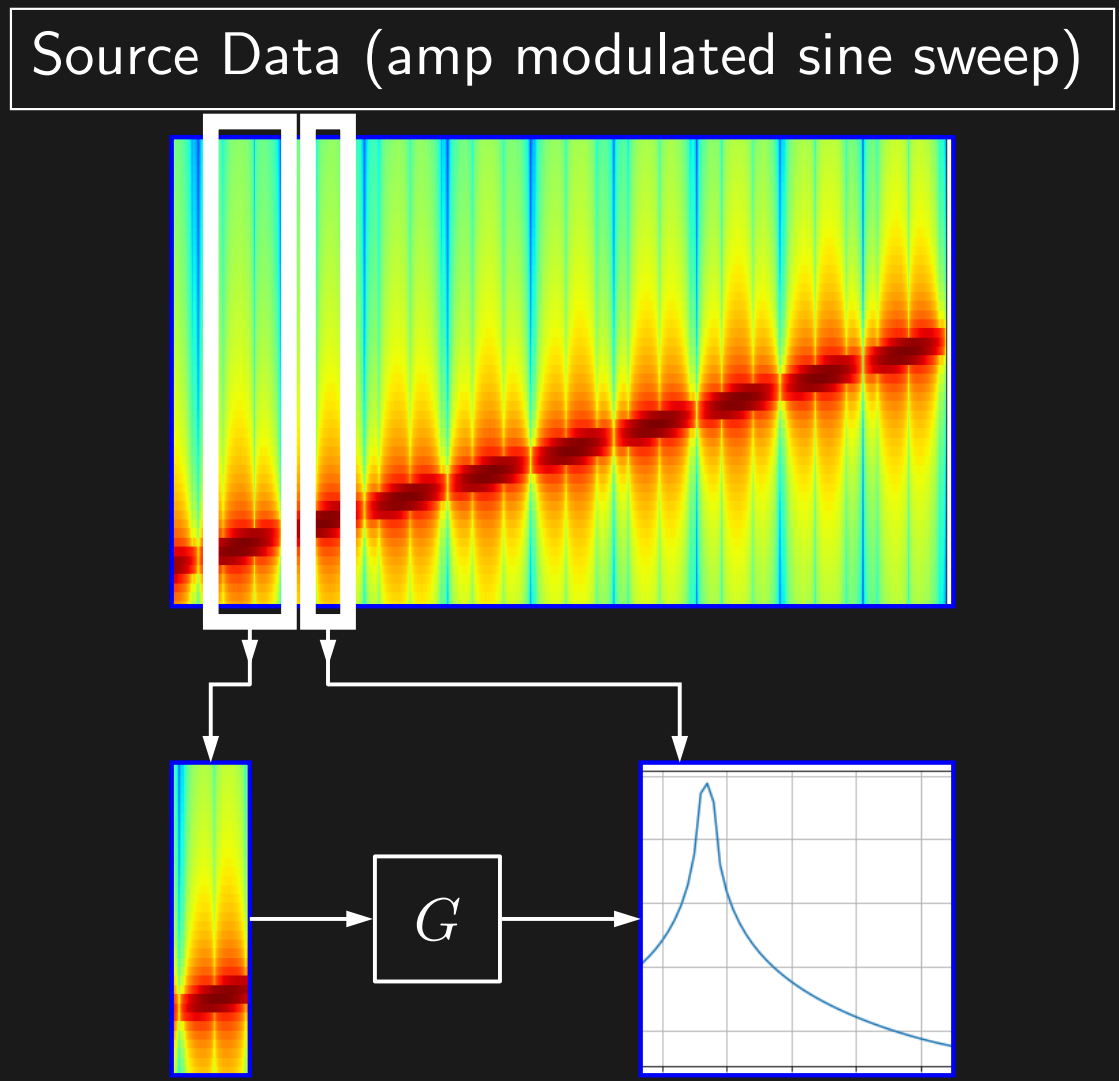
- Create an expressive system that can be used by artists in the field of experimental electronic music
- Use AI to autgenerate content but give control to the artist
- achieve aesthetical results that are directly useful/interesting on their own
- exploit neural network specific artifacts as novel timbres
- Create an architecture with the possibility of real time execution in mind

A Network, G is trained to predict Short-Time Fourier Transform (STFT) Frames, $X[n]$ given k number of past STFT frames so that it ideally satisfies:

$$G\{X[m-1,\omega],...,X[m-k,\omega]\} = X[m,\omega] \tag{1}$$

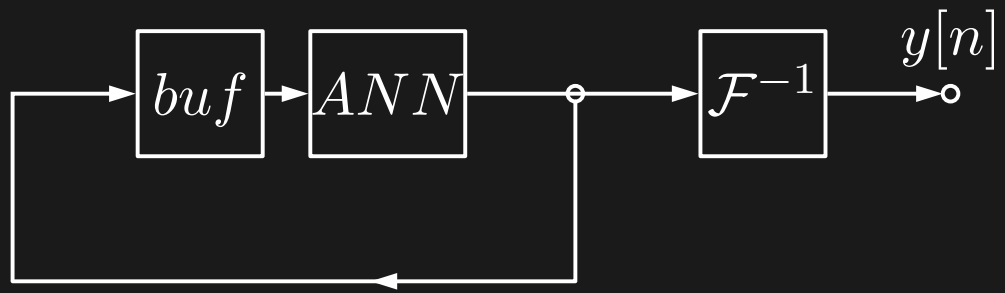
The training data for the network can be chosen at will and has great influence on the aesthetical result.

Example



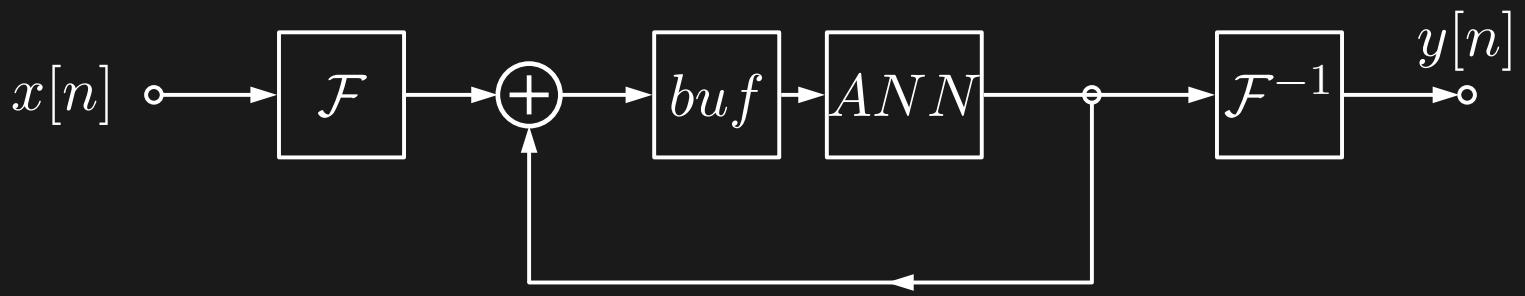
Architecture

Theoretically, such a network should be able to completely reproduce the training data by itself if fed back its buffered output and the buffer being initialized with the beginning of the training set:

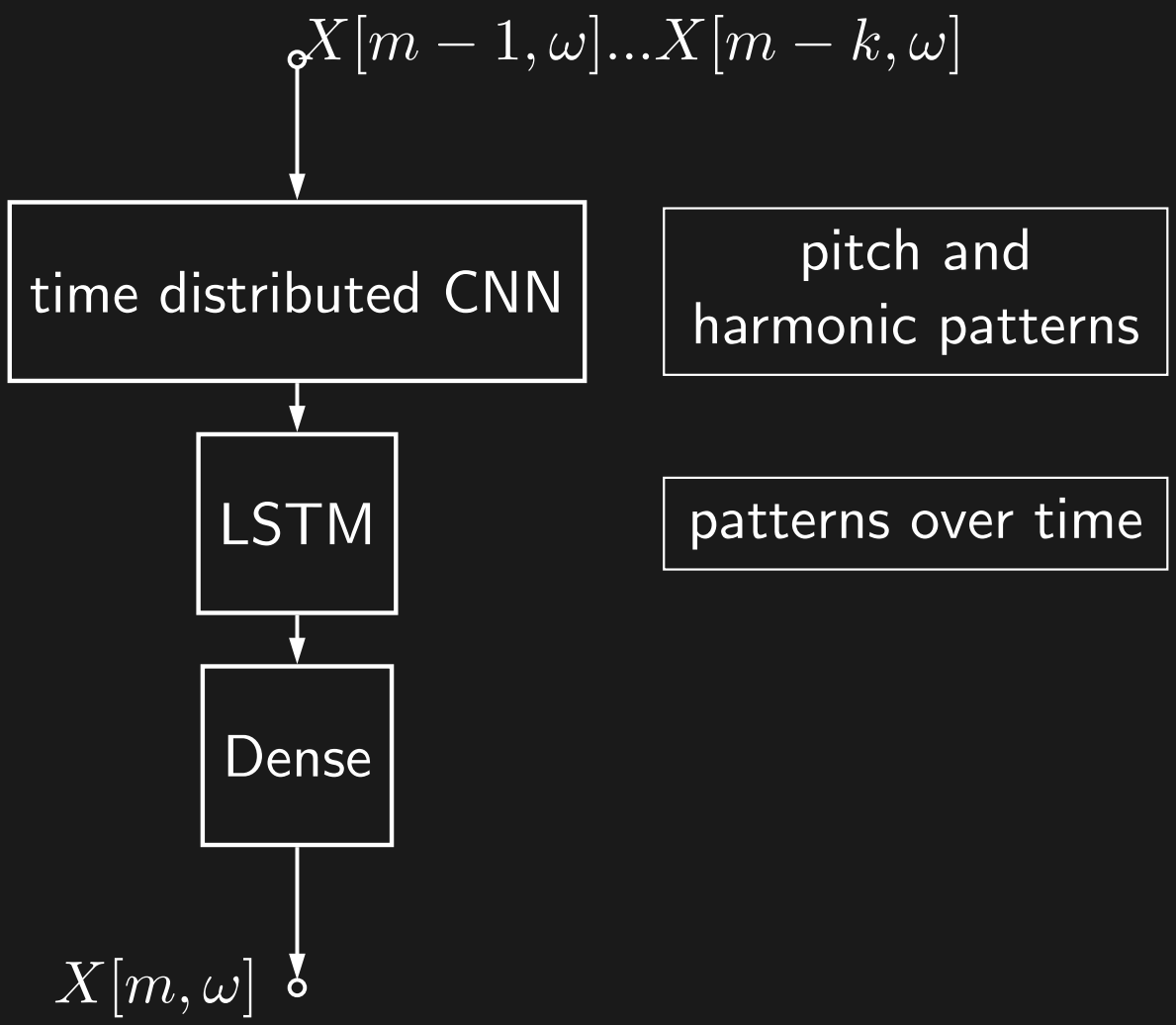


Architecture

This brings the opportunity to feed in an additional fourier transformed time domain signal which can be used to control the networks tendency which parts of the trainingset to reproduce.



ML Architecture



Pre-processing

The chosen dataset containing a monophonic time domain audiosignal is first trasformed via STFT:

$$STFT\{x[n]\}(m,\omega) \equiv X(m,\omega) = \sum_{n=0}^{N-1} x[n]w[n-m]e^{-j\omega n} \quad (2)$$

Pre-processing

The obtained frames are further processed to facilitate the learining process:

- The complex valued Frames are converted to amplitude phase pairs, $\rho(m, \omega)$ and $\phi(m, \omega)$
- The amplitude values are log_{10} compressed
- Phase values of bins whose amplitude value is below a given threshold are omitted.
- Both amplitude and phase data are normalized to $(0, 1)$
- At last, the data is split up to provide input/output data for the learining process

Pre-processing

The final input to the network could look like this:

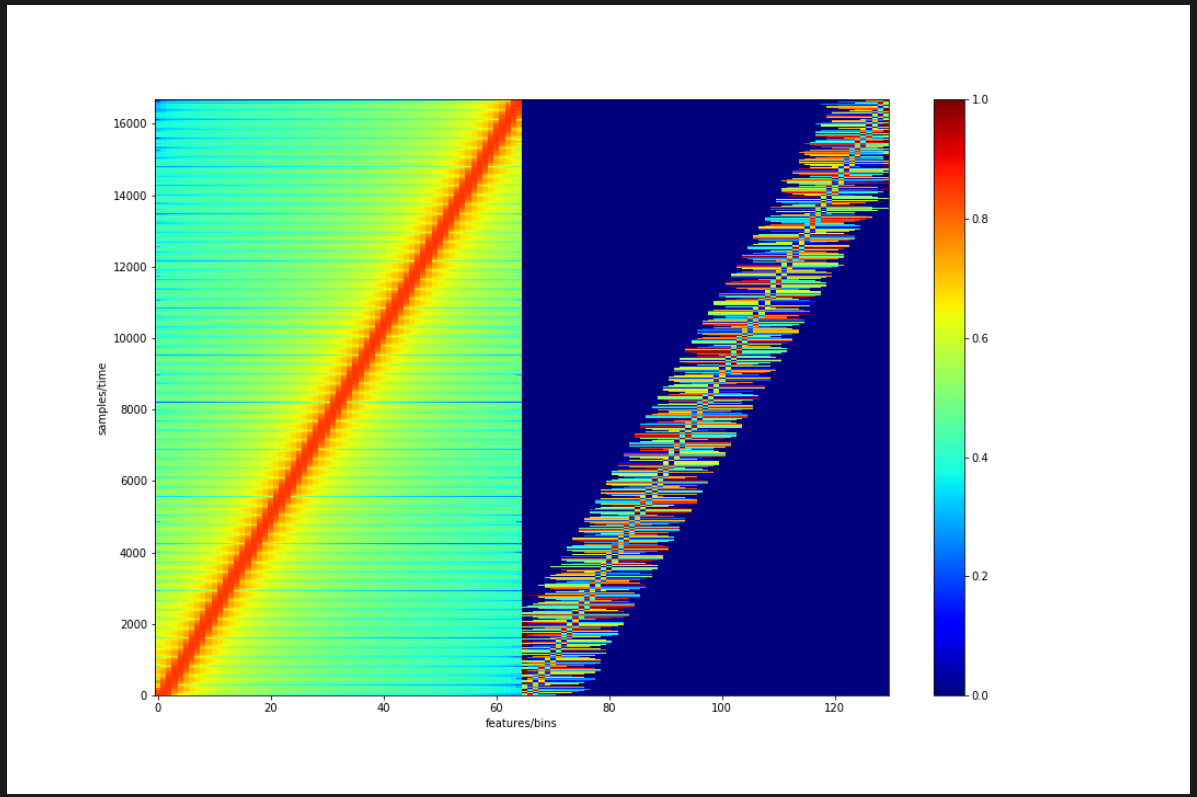
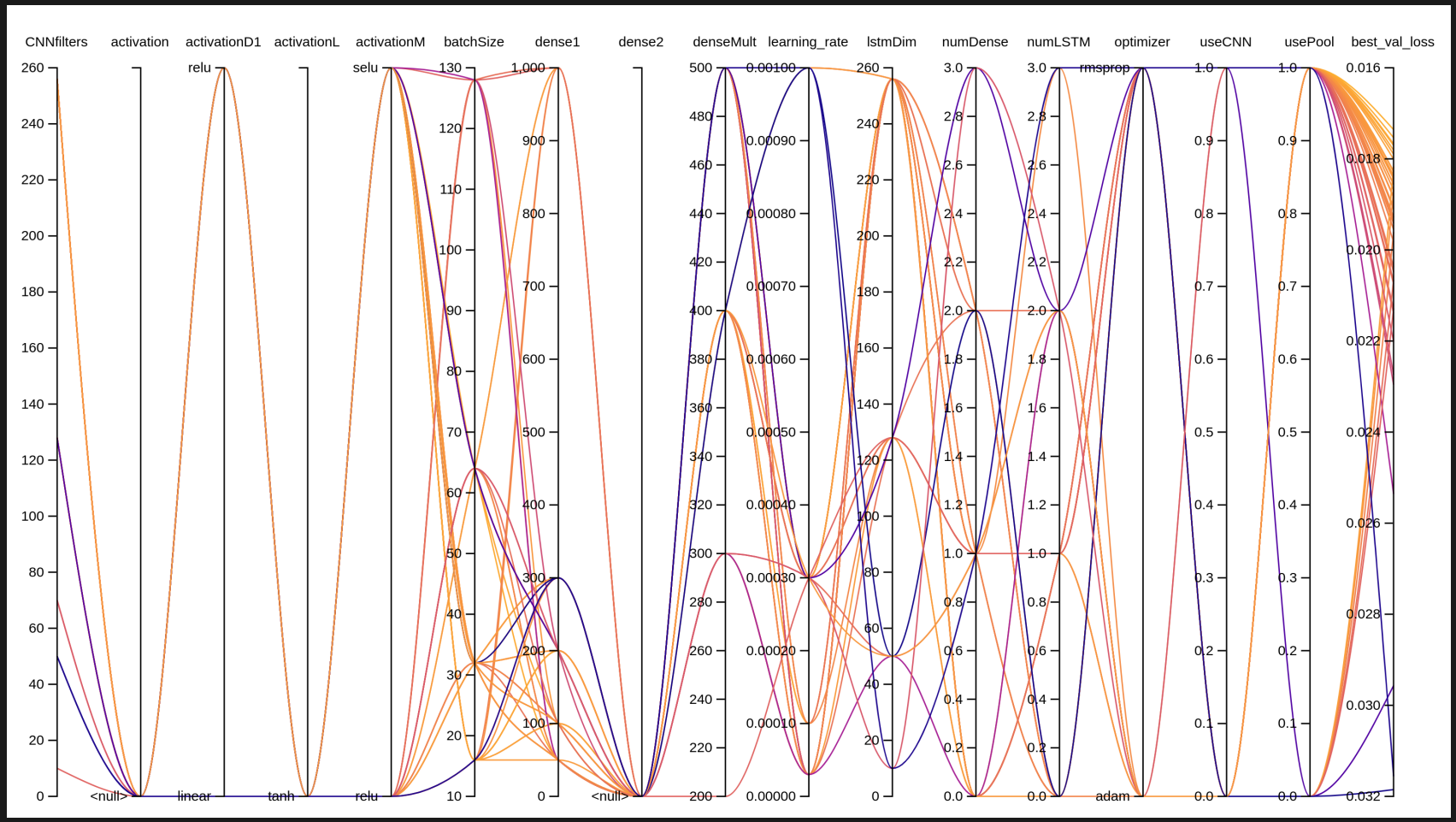


Figure: Training Data example. Amplitude and phase values concatenated. Sine sine sweep was used as input data.

Pre-processing

To speed up learning and increase robustness, phase values can also be omitted completely and generated by input signals or noise in sound generation phase after training.

Model Selection



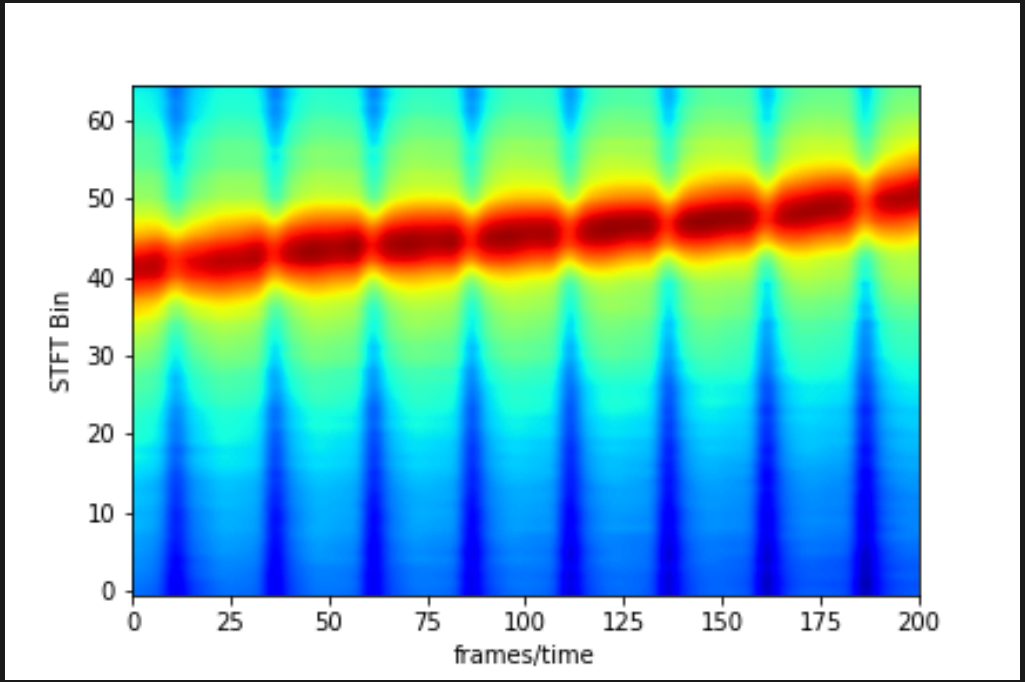


Figure: feedback prediction of a model trained with a modulated sine sweep.

Execution Speed

For a high performance Consumer Grade PC, Sampling rate 18kHz: around 15ms per 1024 bin FFT Frame. This means around 270 samples @ 18 kHz(for model prediction only)

A numer of augmentations to the presentaed sytem can be explored since the proposal is very general:

- Other architectures of the ANN
- Explore different encodings for different aethetical results and arttifacts
- Explore 'Convolutional Reverb'-like use cases
- Further explore different parametrozations (FFT Window length, number of input frames etc.)
- Robustness
- Real time implementation