Audio Track Accompaniment

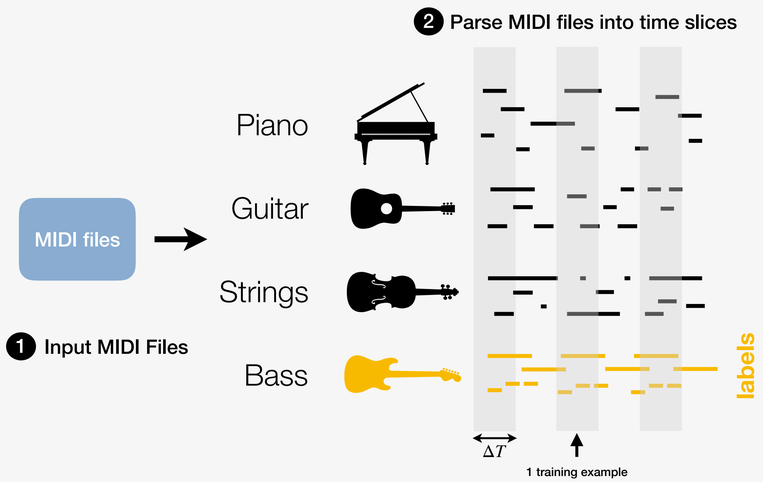
Marion Lepert and Jérôme Nowak {lepertm, nowak}@stanford.edu

Overview

- Music composition, improvisation and accompaniment are creative acts that are challenging to capture algorithmically.
- **Goal:** Given the audio tracks of three instruments as input, **generate an accompanying audio track** for a fourth instrument
- **Approach**: We formulated our problem as a **classification** problem, where each class is one of 128 possible note pitches.

Dataset

- We used the Lakh Dataset, which has over 170,000K MIDI files, and selected a subset of those files that have guitar, piano, strings, and bass tracks.



3 Custom HDF5 dataloader in pytorch

	Piano			Guitar			Strings			Bass		
Training Examples	Note 1	..	Note 128	Note 1	...	Note 128	Note 1	...	Note 128	Note 1	...	Note 128
i = 1	1		0	0		1	0		0	1		0
i = 2	1		0	1		0	1		1	0		0
i = n	0		0	0		0	1		1	1		1

For each time interval: 1 = note played 0 = note not played

Methods

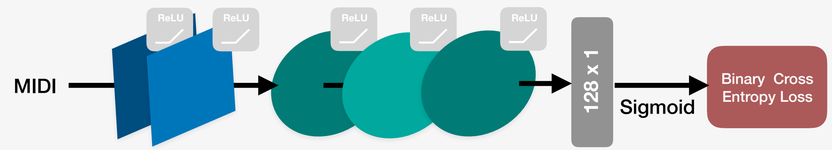
SVM

- As a simple baseline, we started with SVM. We compared performance on a linear and RBF kernel.
- Optimization Problem

$$\min_{\gamma,w,b} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\zeta_i \quad \text{s.t.}$$

$$y^{(i)}(w^T x^{(i)} + b) \geq 1 - \zeta_i, \quad i = 1,...,n, \ \zeta_i \geq 0, \ i = 1,...,n$$

CNN



- We address class imbalance problem by up-weighting positive examples

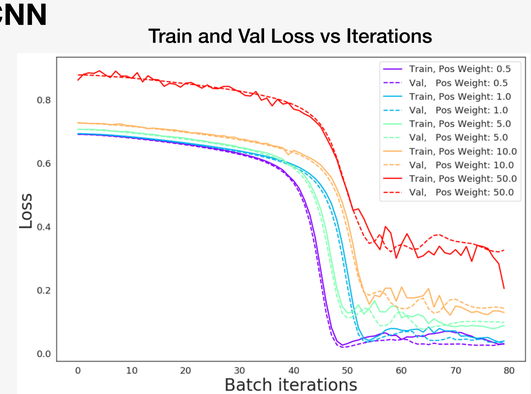
Results

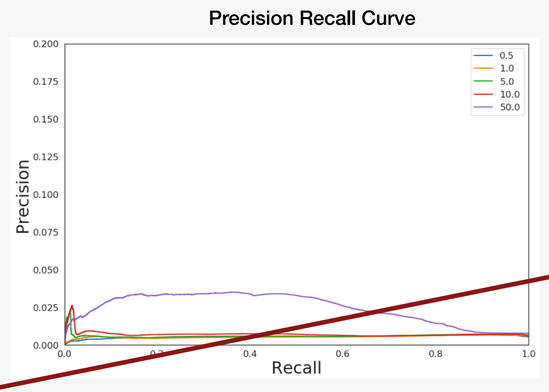
SVM

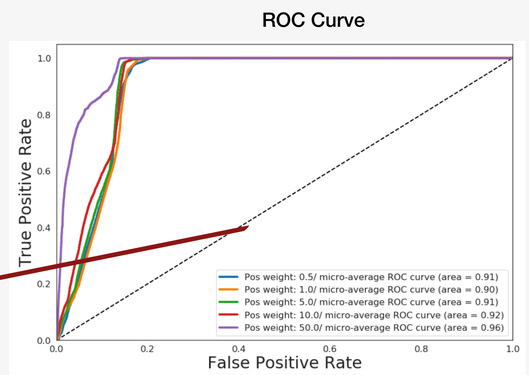
	SVM (linear kernel)	SVM (RBF Kernel)
Accuracy	0.984	0.984
Precision	0.0	0.0
Recall	0.0	0.0

- Both methods suffer from class imbalance problem, even with the use of up-weighting in the neural network loss function
- Our CNN implementation performs slightly better than SVM

CNN







Future Work

- Synthesize MIDI files to audio files and evaluate music with a Turing Test
- Compare CNN performance to RNN and LSTM architectures and reframe the problem as seq-to-seq instead of classification