PianoNet: Deep Neural Network for Automatic Music Transcription

Susan Chang & Maddie Wang
Undergraduates
{schang92@stanford.edu, maddiew@stanford.edu}

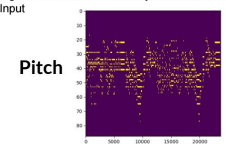
Introduction

Transcribing music from a recording is a time-consuming task that requires extensive knowledge on music. In our project, we offer a hyperparameter study of a deep learning algorithm that can automatically convert a piano MP3 recording into notes. To do so, we use a fully-connected deep neural network approach (DNN).

Prediction

Inputs: Constant-Q Transform of raw audio in matrix form
Outputs: Predicted pitches on piano scale

Figure 1: Predicted Pitches by Frame of Audio Input



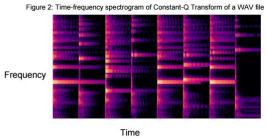
Dataset

- MIDI Aligned Piano Sounds (MAPS) dataset
  - ~31 GB of MIDI-annotated piano recordings. Each **.WAV** recording is paired with aligned **MIDI file** & **txt file** containing off/onset times of notes & pitch (ground truth)
  - 270 classical piano pieces
- Train/dev/test -> 0.78/0.11/0.11
- Train: over 3 million samples
- Mini-batch size of 100
- Dev & Test: over 40000 samples each

Features

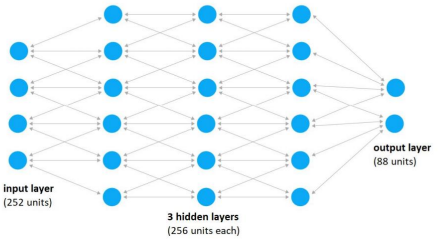
- **Preprocessed** raw WAV files using Constant Q Transform (CQT) over 7 octaves with 36 frequency bins per octave. Meaning, we have 252 total frequency bins per frame, which are derived features.
- CQT is like the Fourier transform except with geometrically spaced center frequencies The frequency of musical notes are in geometric progression, so this makes makes extracting CQT features more suitable for automatic music transcription.

Figure 2: Time-frequency spectrogram of Constant-Q Transform of a WAV file



Model (DNN)

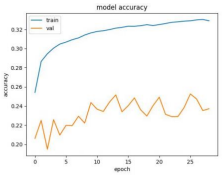
- **Input layer**: CQT preprocessed data
- **Hidden layers**: ReLU activation.
- **Output layer:** sigmoid activation function. 88 units to represent 88 possible pitches of a piano keyboard
- To prevent overfitting, dropout and early stopping were used. The loss function used is the mean squared error between the ground truth labels vector and the predicted labels for each frame.

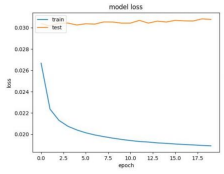


Results

Optimization	Dropout	F-measure	Accuracy
Adam	0.2	69.1	52.7
Adagrad	0.2	64.5	47.6
AdaMax	0.2	69.2	52.9
RMSProp	0.2	66.8	50.1

Dropout	F-measure	Accuracy
0.2	67.1	50.5
0.3	65.5	48.7
0.4	65.6	48.8





Discussion

- Confirmed that Adam performs better than RMSProp optimization.
- AdaMax optimizer gave us better accuracy and f-scores.
- 0.2 dropout rate gave us better accuracy and f-scores than the 0.3 dropout rate for DNN that was suggested in other DNN implementations, but 0.4 dropout rate gave slight improvement.
- Accuracy curve suggests possible overfitting. Prevent future overfitting by removing duplicate music pieces in validation set.

Future work

- Run a parameter study on the convolutional neural network (CNN) architecture and compare its performance to our study on this DNN architecture.
- Augment data such and add noise/deformations to make classifiers more robust.
- Try the RNN approach for music transcription
  - Would be interesting observe the results of using memory units and learning longer note patterns.