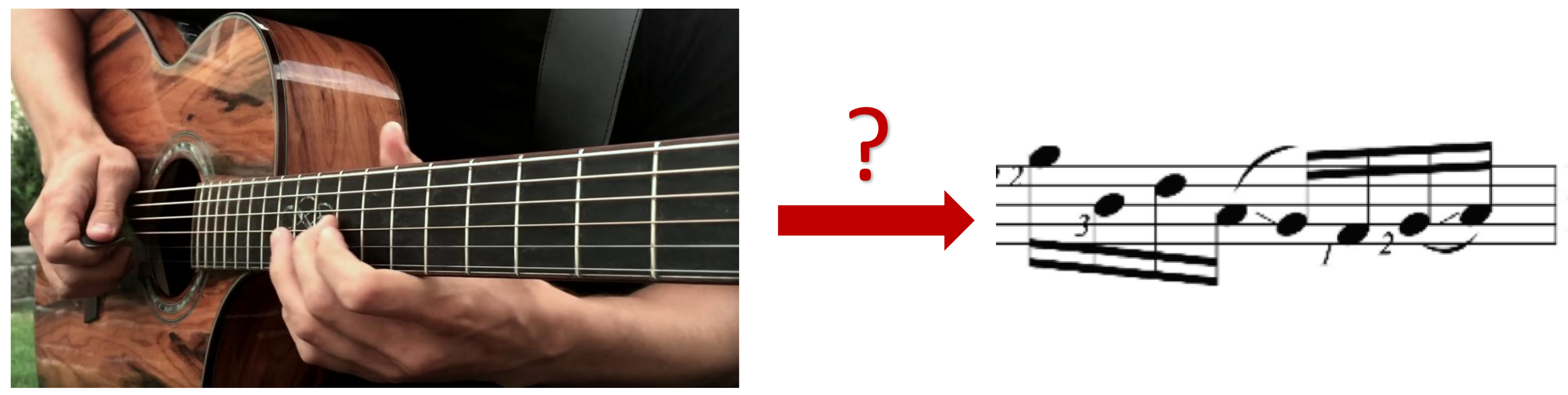
Music Transcription Using Deep Learning

Luoqi Li, Isabella Ni, Liang Yang

liluoqi@stanford.edu; nizhongy@stanford.edu; lyang6@stanford.edu

Motivation



Challenges: Music transcription, based on solid professional knowledge and experiences, cannot be programmed with a certain set of rules directly.

Purposes: This project is going to investigate applying deep learning methods (DNN and LSTM) to music transcription.

Data

Data : 270 pieces from MIDI Aligned Piano Sounds (MAPS) [1]. 60% for training, 20% for validation and 20% for test.

Formats: Audio files (.wav); Ground-truth onset/offset time and pitch for each note (.txt)

Labels: Multi-labeled (88 labels for 88 piano keys) one-hot encoded

Down-sampling: 44.1kHz to 16kHz

Normalization: Training mean was subtracted from 3 sets

Features

Transform: The audio files (.wav) are transformed into spectrograms by Constant Q transform (CQT)

CQT parameters: 7 octaves with 36 bins per octave and a hop size of 512 samples.

Number of features: 252 features per frame

Data matrix:

DNN: $number\ of\ frames \times number\ of\ features$

LSTM: $\frac{number\ of\ frames}{100} \times 100 \times number\ of\ features$

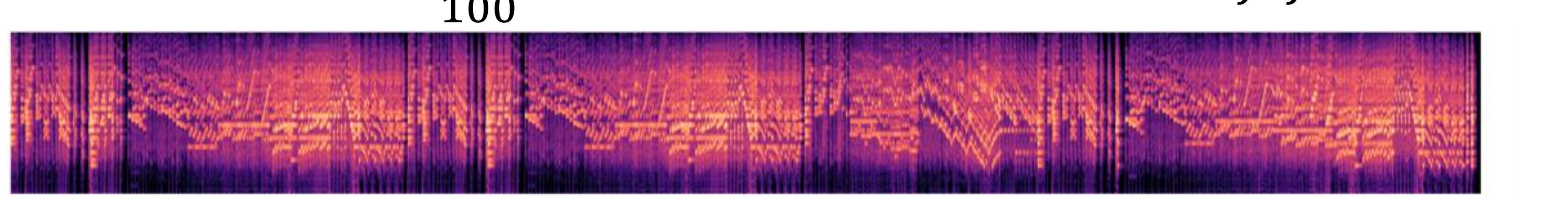


Figure 1. Spectrogram

Models

Compared neural networks: DNN and LSTM [2][3]

Implementation: Keras with Tensorflow backend

Loss function: Binary cross entropy

Activation functions: Sigmoid for output layers; relu (DNN) and tanh (LSTM) for hidden layers

# of layers and units: 3 hidden layers; 256 units per hidden layer; 252 units for input layer; 88 units for output layer

Optimization: Adam optimizer

Strategies to avoid over-fitting: Early stop and dropout

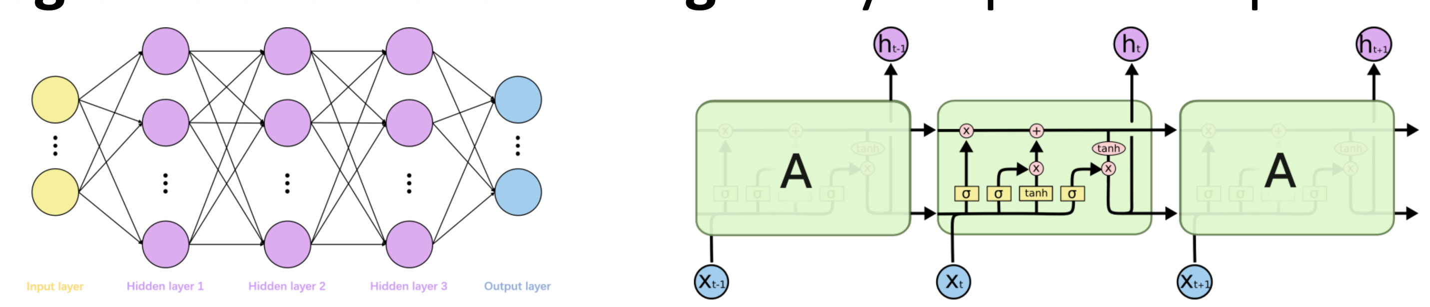


Figure 2. DNN (left) and LSTM (right) architecture illustration

Results

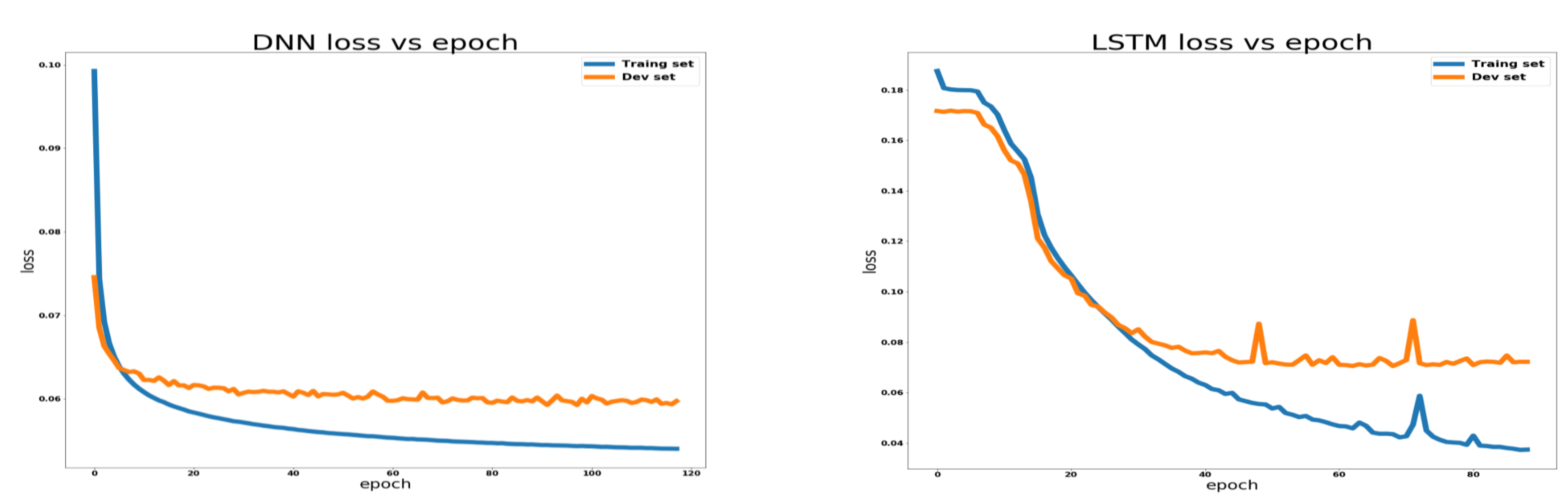


Figure 3. Cross entropy losses for training and validation sets

Performance measures:

$$\text{Precision}(P) = \sum_{t=0}^{T} \frac{TruePositives(t)}{TruePositives(t) + FalsePositives(t)} \qquad \text{Recall}(R) = \sum_{t=0}^{T} \frac{TruePositives(t)}{TruePositives(t) + FalseNegatives(t)}$$

$$\text{Accuracy}(A) = \sum_{t=0}^{T} \frac{TruePositives(t)}{TruePositives(t) + FalsePositives(t) + FalseNegatives(t)} \qquad \text{F}-\text{meature}(F) = \sum_{t=0}^{T} \frac{2PR}{P+R}$$

DNN	0%	10%	15%	20%	25%	30%
F-measure	74.223	**77.581**	77.489	77.491	77.301	76.817
Recall	0.686	**0.716**	0.712	0.706	0.707	0.701
Accuracy	59.012	**63.373**	63.250	63.254	63.001	62.360

LSTM	0%	10%	15%	20%	25%	30%
F-measure	65.638	68.431	69.476	74.074	**75.586**	74.449
Recall	0.589	0.627	0.642	0.688	**0.711**	0.691
Accuracy	48.852	52.072	53.229	58.824	**60.754**	59.298

Table 1. DNN and LSTM performance with different dropout rates

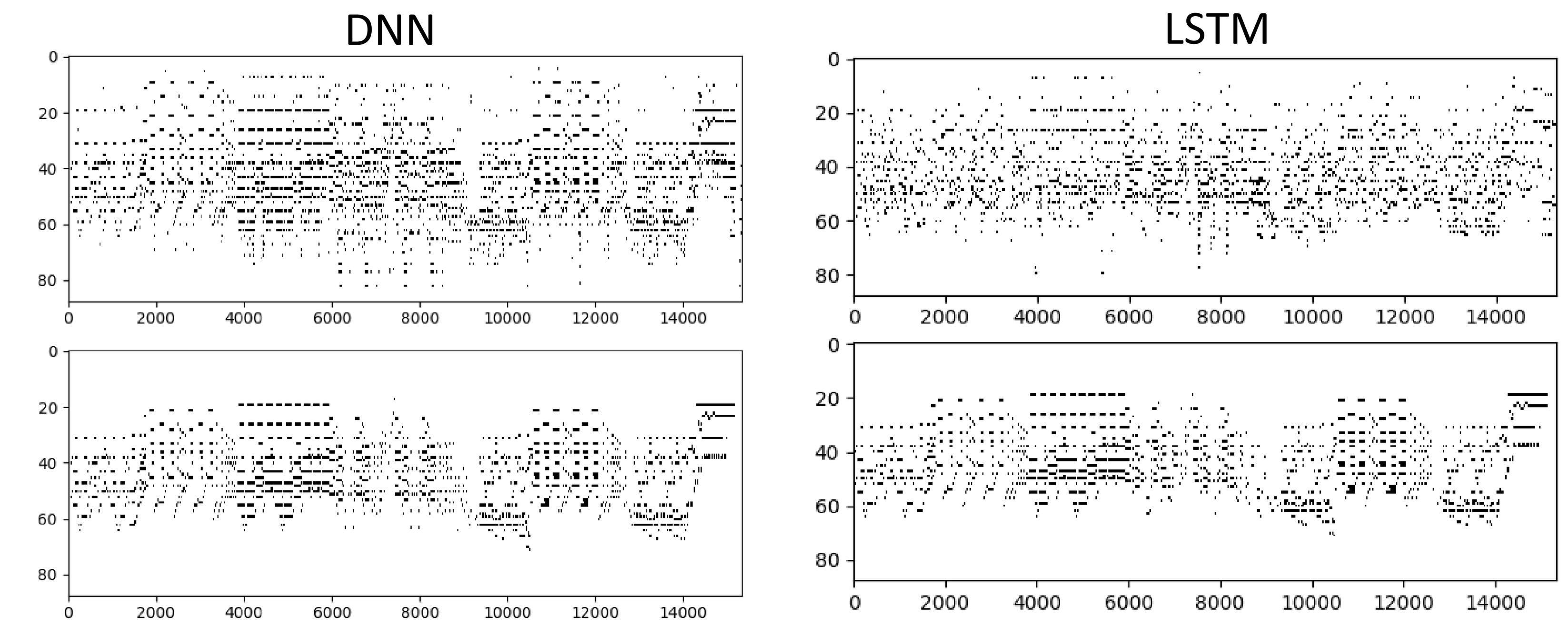


Figure 4. DNN /LSTM predictions (top) and ground truth (bottom)

Figure 5. An excerpt from the predicted music scores

Discussion

Some parts of the predicted music scores are playable. The prediction accuracy is promising, given that our data set is small and the neural network is not very deep. LSTM was supposed to perform better than DNN, but we obtained the opposite results. Compared to DNN, LSTM has a smaller training loss and a larger validation loss (Fig. 3). Also, LSTM's performance improved when increasing dropout rates to 25% (Table 1). We think the reason that LSTM performs worse is overfitting. Tests for LSTM with less hidden layers, units, and larger dropout rates are needed in the future.

Future

1. Try different preprocessing procedures
2. Test different neural network parameters
3. Test more data

References

[1] V. Emiya, N. Bertin, B. David and R. Badeau, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Transactions on Audio, Speech and Language Processing*, (to be published). Available: http://www.tsi.telecom-paristech.fr/aao/en/2010/07/08/maps-database-a-piano-database-for-multipitch-estimation-and-automatic-transcription-of-music

[2] D. G. Morin, "Deep neural networks for piano music transcription," Jun. 2017. Available: https://github.com/diegomorin8/Deep-Neural-Networks-for-Piano-Music-Transcription.

[3] S. Sigtia, E. Benetos and S. Dixon, "An end-to-end neural network for polyphonic piano music transcription," *IEEE/ACM Trans. Audio Speech Lang. Process.*, 24, 927–939, 2016. Available: https://arxiv.org/abs/1508.01774.