MIREX2024: Piano Transcription using Feature-wise Linear Modulation

Qi Wang, Mingkuan Liu, Shaoxin Yang
Beijing University of Technology, China
wangqi91@bjut.edu.cn

Abstract—**This submission presents a piano transcription system that utilizes the feature-wise linear modulation (FiLM). The system is designed with the conventional convolutional and recurrent networks as the FiLM generator. The fusion of features from different tasks is also implemented using the FiLM. The FiLM-based model has only 1.2M trainable parameters.**

I. Introduction

The purpose of piano transcription is to transform piano music to note events with pitch, onset, offset and other music symbols. With the development of deep learning, the neural networks (NNs) have been widely used in piano transcription. The multi-task framework achieved superior performance in transcription, such as the onsets and frames (OAF) model [1], [2]. In the multi-task framework, frame-wise pitch estimation, note-level onset and offset detection, and velocity estimation are the main transcribing subtasks. In this work, we implement a multi-task piano transcription system with feature-wise linear modulation (FiLM).

II. Feature-wise Linear Modulation

The FiLM layers were first proposed in [3] as a general-purpose conditioning method for visual reasoning tasks. Specifically, the modulation operation of the FiLM layers is shown as:

$$\text{FiLM}(x|\gamma,\beta) = \gamma \odot x + \beta \tag{1}$$

where x is the input feature, γ and β are the context features generated from the FiLM generator. $\odot$ refers to Hadamard product.

III. System architecture

The multitask piano transcription task in this study aims to transcribe the T-frame input spectrogram into multiple outputs. The outputs are the T-frame and K-pitch piano rolls of transcription subtasks. These subtasks include frame-level pitch estimation, onset detection, offset detection, and velocity estimation.

The overall schematic diagram of the proposed multitask piano transcription model is shown in fig:overall. There are mainly three branches in the proposed multitask piano transcription model, which are the onset, the velocity and the frame & offset branches.

The system comprises the input layer, the FiLM-based layers, and the output layer. There are a total of four FiLM-based layers stacked in every subtask branch. In each FiLM-based module, the feature map is firstly passed by a convolution layer. In the onset branch, these prepared γ and β features are detached and sent to the velocity and the frame branches. Then the velocity and the frame branches receive and concatenate these γ and β features to the parallel positions. After that, all the features are rearranged with temporal information through the LSTMs. Finally, the input is linear modulated by the generated frequency-time feature. To generate proper shape of the model output, there is a multilayer perceptron (MLP) connected to the FiLM-based layers in each branch. All the output piano rolls are then scaled by a sigmoid function except for the velocity branch as the final model output.

The losses for the frame, onset, and offset are calculated with binary cross-entropy. The loss for the velocity is a 128-category cross-entropy loss. Specifically, the losses are:

$$l_{\text{onset}} = \sum_{k=1}^{K}\sum_{t=1}^{T} l_{\text{BCE}}(I_{\text{onset}}(t,k), P_{\text{onset}}(t,k)) \tag{2}$$

$$l_{\text{frame}} = \sum_{k=1}^{K}\sum_{t=1}^{T} l_{\text{BCE}}(I_{\text{frame}}(t,k), P_{\text{frame}}(t,k)) \tag{3}$$

$$l_{\text{offset}} = \sum_{k=1}^{K}\sum_{t=1}^{T} l_{\text{BCE}}(I_{\text{offset}}(t,k), P_{\text{offset}}(t,k)) \tag{4}$$

$$l_{\text{velocity}} = \sum_{k=1}^{K}\sum_{t=1}^{T} l_{\text{CCE}}(I_{\text{velocity}}(t,k), P_{\text{velocity}}(t,k)) \tag{5}$$

where l_{BCE} is the binary cross-entropy loss function, l_{CCE} is the categorical cross-entropy loss function, I and P are the ground-truth and model-predicted values, respectively. The total loss function is calculated by:

$$l = l_{\text{onset}} + l_{\text{frame}} + l_{\text{offset}} + l_{\text{velocity}} \tag{6}$$

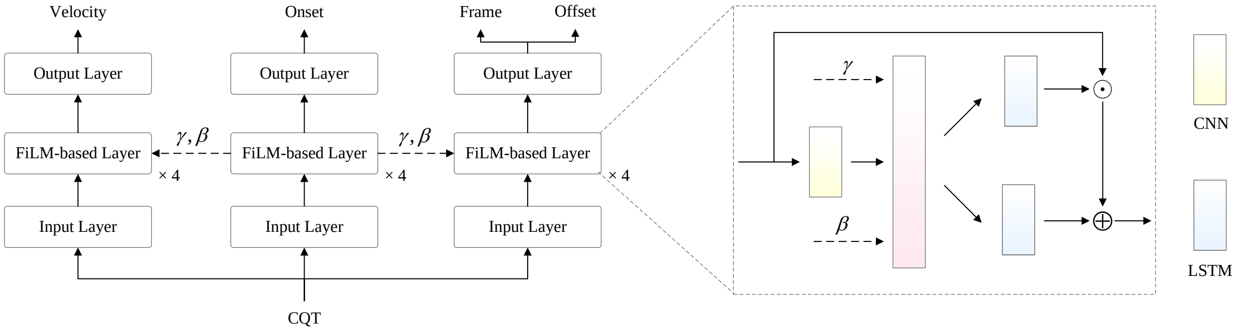

Fig. 1. Architecture of the piano transcription system.

IV. Dataset

The MAESTRO dataset [2] contains about 200 hours of high-quality realistic recordings and MIDI files. We use the provided train, validation, and test splits in the MAESTRO V3.0.0 dataset.

The model trained on MAESTRO V3.0.0 is evaluated on the MAPS dataset [4]. We use 60 audio recordings of realistic piano pieces in the "ENSTDkAm" and "ENSTDkCl" as the evaluation test split.

The model trained on MAESTRO V3.0.0 is also evaluated on the SMD dataset [5]. The dataset contains 50 recordings. Similar to Maestro dataset, the SMD dataset was created by recording human performance on a Yamaha Disklavier.

V. Preprocessing

The *librosa* [6] library is used to preprocess the piano audios. The audio is converted to mono and resampled to 16kHz. Each piano audio is transformed to CQT spectra with a Hanning window, minimum frequency of 27.5Hz (A0), a 20-millisecond frame hop length, every 48 bins within an octave and a total 352 bins. The *pretty_midi* [7] library is used to generate the piano rolls. We also consider the influence of the sustain pedal on extending notes duration. Each piano note is extended until the pedal is released (pedal value is less than 64 in MIDI files) or the same note is pressed again.

VI. Training

The piano transcription model is trained on MAESTRO V3.0.0 training set without data augmentation. There are a total of 1.2M trainable parameters. The training is optimized with the Adam [8] optimizer. We use a learning rate of 0.0005 and a batch size of 10. The best model is selected based on the F1 scores in the validation stage.

VII. Transcribing

To obtain the note events, the heuristic note event decoding method [1] is utilized to decode the note events from the output post-probability piano rolls. All the thresholds are decided based on the result of the validation set. The *pretty_midi* library is used to generate the MIDI files.

References

[1] C. Hawthorne, E. Elsen, J. Song, *et al.*, "Onsets and frames: Dual-objective piano transcription," in *19th International Society for Music Information Retrieval Conference, ISMIR 2018, September 23, 2018 - September 27, 2018*, 2018, pp. 50–57.

[2] C. Hawthorne, A. Stasyuk, A. Roberts, *et al.*, "Enabling factorized piano music modeling and generation with the Maestro dataset," in *7th International Conference on Learning Representations, ICLR 2019, May 6, 2019 - May 9, 2019*, 2019.

[3] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018, pp. 3942–3951.

[4] V. Emiya, R. Badeau, and B. David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1643–1654, 2010.

[5] M. Meinard, K. Verena, B. Wolfgang, and A.-M. Vlora, "Saarland music data (smd)," in *Late-Breaking and Demo Session of the 12th Conference of International Society for Music Information Retrieval (ISMIR)*, 2011.

[6] B. McFee, C. Raffel, D. Liang, *et al.*, "Librosa: Audio and Music Signal Analysis in Python," in *Proceedings of the 14th Python in Science Conference*, K. Huff and J. Bergstra, Eds., 2015, pp. 18–24.

[7] Colin Raffel and Daniel P. W. Ellis., "Intuitive analysis, creation and manipulation of MIDI data with pretty_midi," in *Proceedings of the 15th Conference of International Society for Music Information Retrieval (ISMIR)*, 2014.

[8] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.