Towards audio based score following

Philipp Ostermaier
Audiocommunikations
Technical University Berlin
Berlin, Germany
p.ostermaier@campus.tu-berlin.de

Maximilian Wiedenmann
Audiocommunications
Technical University Berlin
Berlin, Germany
m.wiedenmann@campus.tu-berlin.de

Anaïs Bayani
Computer Science
Technical University Berlin
Berlin, Germany
bayani.mianodab@campus.tu-berlin.de

***Abstract*—This paper introduces a deep learning approach to score following using dual inputs of audio spectrograms which represent reference audio of several seconds length as well as short snippets of individual onsets' temporal surrounding. The neural network employed consists of a dual stack of layers with several steps of convolution and a concatenation layer to combine the stacks. A "soft target" vector relating to the reference spectrogram frames represents discretized timestamps and serves as ground truth for the training process. The network is trained on the MAPS dataset as well as the Giant Midi Piano Dataset. The results for monophonic audio input presented in this paper underline the validity of the approach.**

I. Introduction

Score following has been an activate research field in Music Information Retrieval (MIR) going back to the 1980s. Various methods have been developed to tackle the problem, mostly relying on symbolic audio input data like midi as a noise free input. Among the most established methods are machine learning methods like Dynamic Time Warping (DTW) [1] and Hidden Markov Models (HMM) [2]. In DTW the mostly likely score position is determined by mapping the symbolic audio data to the score of a given piece in a cost matrix, which is then traversed by the DTW algorithm establishing a lowest cost path through this matrix. Hidden Markov chains are stochastic methods which predict for a given reference and an observed sequence of audio events (in this context also referred to as emissions) the most likely state the and thus the current position of the performance. Generally these methods require a considerable understanding of the matter to be able to describe underlying features so that good results can be achieved, while their main advantage being requiring little computational overhead.

With rising availability of compute resources and accelerating advances in hardware and frameworks since the early 2000s, deep learning methods have played an increasing role also in the domain of score following.

This project represents a modified approach to, "Towards score following in sheet music images", by M. Dorfer, A. Arzt, G. Widmer from 2016 [4]. In their work they use a dual stack deep neural net to directly relate position of note onsets in pixel-images of piano sheet music to onset events in spectrogram snippets of piano performance audio. The system described allows to predict the horizontal pixel position within the sheet music. As the sheet music notation is usually not evenly quantized for graphical and legibility reasons, this pixel position is not linearly related to time. Hence, as this work is meant to serve as a step towards an automatic accompaniment system for pianists, the sheet music is replaced by a reference spectrogram.

II. Methods

A. Deep Neural Network

As mentioned in the introduction, we follow a modified approach first shown in [4], employing a dual-stack deep neural network. As input serve 2 types of spectrograms. A 12 second long reference spectrogram representing several bars of music and a 1.35 second long excerpt snippet surrounding a single onset of interest. This temporal context is crucial in order for the network to be able to learn about the position of a specific onset and not mistake it for a similar pitch note at a different time. This is illustrated in Fig. 1.

The dual-stack neural network for these 2 inputs are composed of convolutional layers, Max Pooling Layers with Drop Out and a final Dense Layers with dropout. They are then concatenated to a single stack with 2 Dense Layers with Dropout and a final Soft-Max layer. 3 versions of varying depth of this setup have been used for testing see Tab. I for the final models the most complex model following the setup in [4] has been used. Due to increased size of the reference spectrogram and snippet spectrogram, out final model has more than 10 Million parameters. (Tab. II)

The target of the network is a 2-hot encoded vector corresponding to individual frames within the reference spectrogram. This is equivalent to a discretisation of time. The 2-hot encoded gives additional insight for evaluation, narrow misses of the right label will be covered by the secondary "bucket". In many cases this could be already sufficient for a score follower, as slight temporal deviations of a few milliseconds are often not perceptible.

B. Datasets

As we are specifically interested in score following for sheet based piano music, we choose 2 datasets for testing which are purpose made, yet provide for different needs. The MAPS dataset has a dedicated monophonic section with chromatic

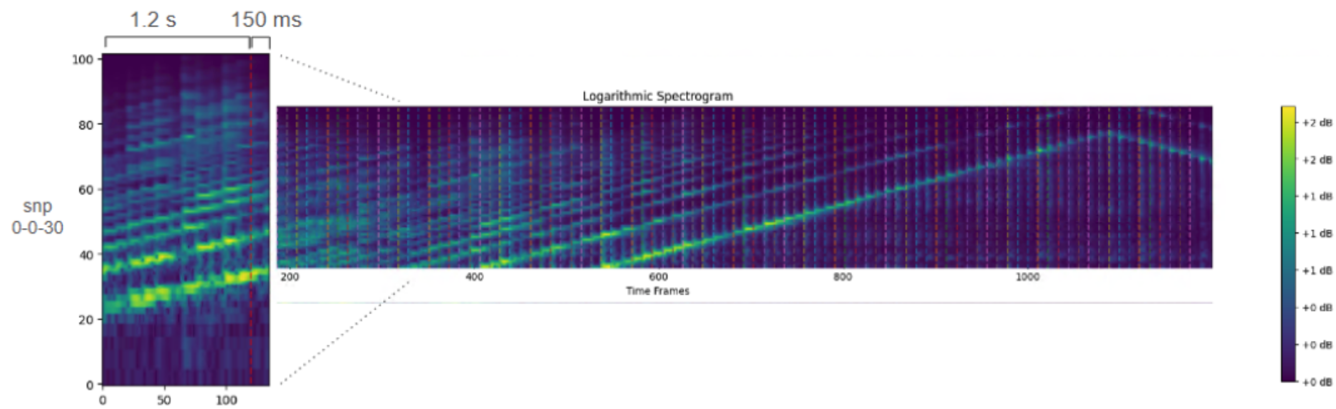

Fig. 1. Reference spectrogram with onsets and snippet with onset

Reference	Snippet
3 x 3 Conv(pad-2) BN + ReLU 2 x 2 MaxPooling Flatten	3 x 3 Conv(pad-2) BN + ReLU 2 x 2 MaxPooling Flatten
Concatenation layer Dense layer + BN + ReLU + Drop-Out(0.15) Dense Layer Softmax	

TABLE I
REDUCED NEURAL NETWORK

Reference	Snippet
3 x 3 Conv(pad-2) BN + ReLU 3 x 3 Conv(pad-2) BN + ReLU 2 x 2 MaxPool + Drop-Out (0.15) 3 x 3 Conv(pad-2) BN + ReLU 2 x 2 MaxPool + Drop-Out (0.15) 3 x 3 Conv(pad-2) BN + ReLU 2 x 2 MaxPool + Drop-Out (0.15) Dense BN + ReLU + Drop-Out(0.3) Flatten	3 x 3 Conv(pad-2) BN + ReLU 3 x 3 Conv(pad-2) BN + ReLU 2 x 2 MaxPool + Drop-Out (0.15) 3 x 3 Conv(pad-2) BN + ReLU 2 x 2 MaxPool + Drop-Out (0.15) 3 x 3 Conv(pad-2) BN + ReLU 2 x 2 MaxPool + Drop-Out (0.15) Dense BN + ReLU + Drop-Out(0.3) Flatten
Concatenation layer Dense layer + BN + ReLU + Drop-Out(0.3) Dense layer + BN + ReLU + Drop-Out(0.3) Softmax	

TABLE II
FULL NEURAL NETWORK

scales and single notes, thus provides very elementary training data, while the Giant Midi Piano dataset provides for polyphonic performances of a large number of classical piano pieces.

In order to create suitable input for our network preprocessing of the datasets from .wav data of varying length to fixed size spectrograms and snippets was necessary. We hereby made sure that our preprocessing routines would produce continuing, overlapping references and snippets for every single onset in a piece. This will be crucial for further evolution steps of the work towards following full pieces of music.

1) MAPS dataset: We use the MAPS (Midi aligned piano sounds) dataset. This dataset contains aligned ground truth of audio, midi and onset and offset data of individual notes. The data is provided with auralisations of several instruments in varying spatial settings, thus provides for different auralisation qualities which can be beneficial for learning performance of deep neural nets.

We focused on the subset ”ISOL” containing chromatic scales covering the full range of the keyboard (CH), isolated notes with a normal articulation (NO), staccato (ST) notes and repeated notes (RE). If the network is able to learn on such a baseline dataset, this will prove basic validity of our approach.

2) GiantMIDI-Piano dataset: This dataset contains a large number of classical piano solo pieces in midi format. Each midi file contains the onset, offset, pitch, and velocity attributes of piano notes as well as the sustain pedal events.

We decided to reduce these polyphonic pieces of music to monophonic by extracting single voices. The frequent pitch changes present a bigger challenge for the training of the network and shall underline its potential usability for actual following of entire musical pieces.

Additional capabilities of our data processor allows for manipulation of notation properties like e.g. the articulations, rests and note length. Further, to cover the entire spectrum of the piano, the pitch of individual notes can be changed. The use of soundfonts allows for auralization of a variety of different instruments with individual piano sound and reverberation characteristics. These tools serve current and future development of the project data augmentation - we employed the auralization with soundfonts in order to create subtle differences in the spectrograms using 1 instrument for

the references while using others for the snippets. The goal is to aid learning of the neural net and support its generalisation capabilities.

III. Results

A. MAPS dataset, tests with reduced model

Initally the reduced model was run on the MAPS chromatic scales (CH) and Staccato notes (ST) subsets. The maximum number of epochs was set to 100. IN case of stagnation for a maximum of 10 epochs an Early Stopping callback function was set to abort training. We evaluated different batch sizes with batch normalisation (BN), drop out, and a max pooling size of (2,2). Then we fixed the batch size to 4 and tried with no batch normalisation or only on the concatenated layer and then only on the special nets. Then we changed the max pooling size to (4,4) (with BN and batch size = 4). Finally, we tested with no dropout. The results can be observed in Tab. III, and Fig. **??** show us the graphs of loss and accuracy for batch size equals to 4.

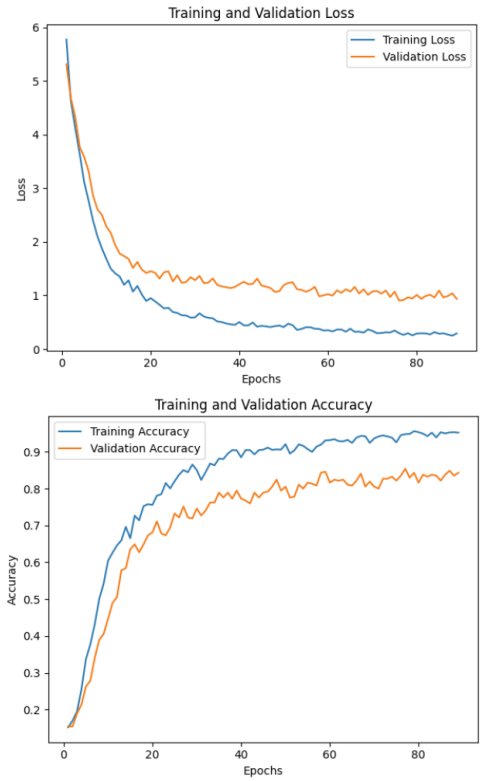

Fig. 2. MAPS CH/ST, batch size = 4 , Training/ Validation, Loss and Accuracy

The results show us that a Max Pooling size of (4,4) is too large to capture the details and the model cannot generalise well. Batch normalisation is necessary on the stacked networks and not particularly on the concatenated layer. Using batch normalisation on the concatenated layer does not change the results much, as the layer is built on a latent representation of the stacked networks that have already been batch normalised. The results are globally the same for different batch sizes, although we can observe a slightly better result with a batch size of 4 (in average 0.03 more test accuracy than the others, and not as much difference between the train and test accuracies.

TABLE III
Results for MAPS dataset, reduced model

	n epochs	Train Accuracy	Test Accuracy
batchSize=4	89	0.94	0.83
batchSize=8	40	0.97	0.81
batchSize=16	35	0.88	0.79
batchSize=32	36	0.99	0.79
No BN	53	0.98	0.77
BN on concatenation	58	0.92	0.73
BN on stacks	100	0.97	0.81
MaxPoolSize=4,4	79	0.90	0.69

The training accuracy, averaging around 0.96, suggests that the model may be overfitting, since the difference between the train and test accuracy is 0.15. The best test accuracy is 0.83, which is a good result for a reduced network. The unique convolutional layer in each stacked dual network is able to capture the particularities of the data.

B. MAPS dataset, tests with full model

For testing the complete ISOL set - auralised with a Boesendorfer 290 Imperial in a reverberation rich church - we used the "full model" II. We employed a slow training strategy with low learning rate of 0.001 and a small batch size of 3. The training took place over more than 500 epochs and could demonstrate a steady improvement over the course of the whole training which is illustrated in figure 3

The bucket-1 and bucket-2 hitrate illustrated in figures 4 and 5 underline the quality of the results. Just a marginal amount of onsets deviate significantly from their intended time, many of the other onsets are within 250 ms of the intended onset time.

For further evaluation of this model we used the smaller CH and ST dataset which was previously employed in the reduced model test. Admittedly, the CH and ST data is also part of the ISOL set, however a different piano was used here (a Concert grand) for auralisation. Also the spatial setting of a studio makes for a less reverberant environment which will influence the spectrograms. We can demonstrate an evaluation accuracy of 66 percent under these conditions.

C. GiantMIDI-Piano dataset

For this even larger dataset, we also used the "full" version of our network. The dataset contains 4 different piano auralisations and was conceived so that different instruments for reference and snippet auralisations are used. The 3 piano pieces used offer more variation in pitch, note lengths and rests, so another uptick in training data complexity in comparison to the MAPS ISOL.

We also here employed a strategy of starting training with a very low learning rate of 0.001 using a Reduce learning

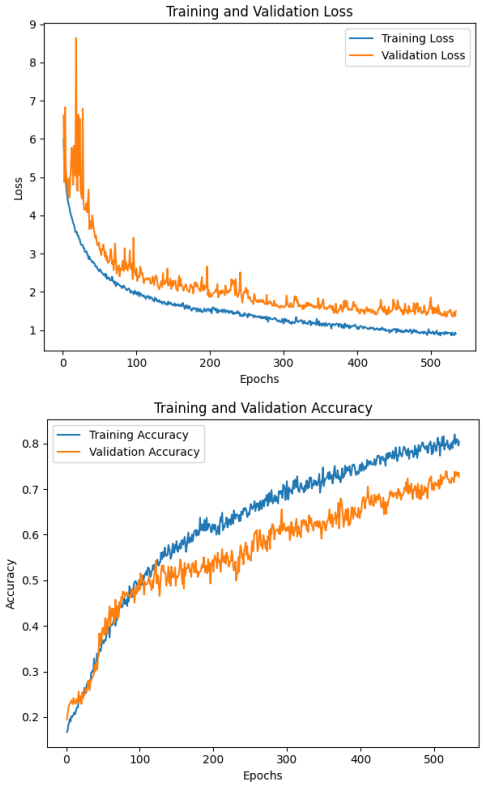

Fig. 3. MAPS ISOL, Training/ Validation, Loss and Accuracy

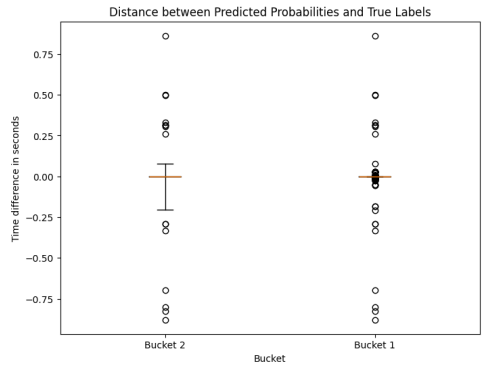

Fig. 4. MAPS ISOL, Distance between Predicted Probabilities and True Labels

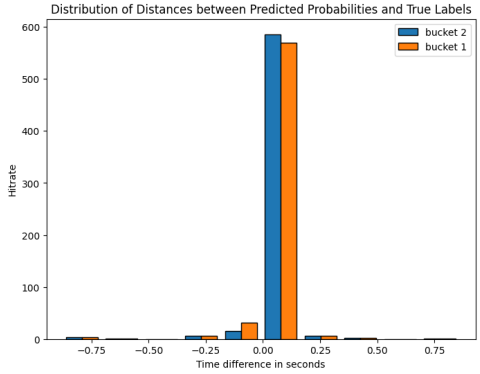

Fig. 5. MAPS ISOL, Distribution of Distances between Predicted Probabilities and True Label

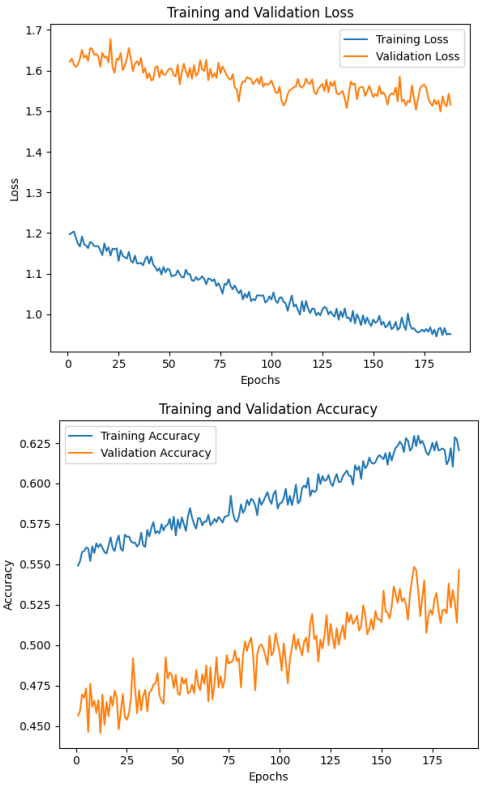

Fig. 6. GMP, Training/ Validation, Loss and Accuracy

rate on plateau callback function. After around 100 epochs we switched to a more aggressive strategy with a batch size of 10 and switching the reduced learning rate callback off. The model showed from that point onwards steady improvement towards the final values shown in figure 6 which it could not further improve.

The results, although not as good as for the MAPS ISOL dataset, are encouraging, we could achieve accuracy of 63 percent for the training data and 54 percent on the test data. The spread on the bucket hits is consequently more pronounced. Delays of 0.5 seconds and more would be noticeable for downstream score following applications.

At best, we obtained a training accuracy of 0.62 with a testing accuracy of 0.53 as can be seen in fig.6. Although it is not a very high accuracy, the model was able to predict half the testing data correctly. Moreover, it seems to have a good bias-variance trade-off, since the training and testing data give close results. Adding more parameters and layers to the model could make it more efficient, but the added complexity could also lead to overfitting. Fig.8 and 7 show us the distance and distribution of distances between the predicted and true buckets. As can be seen, the first bucket tends to be more accurately predicted than the second. While running the model, we observed a discrepancy when the parameters

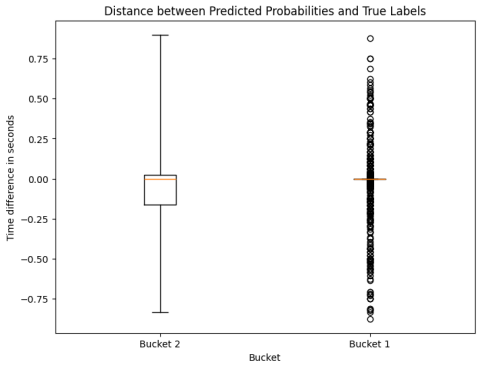

Fig. 7. GMP, Distance between Predicted Probabilities and True Labels

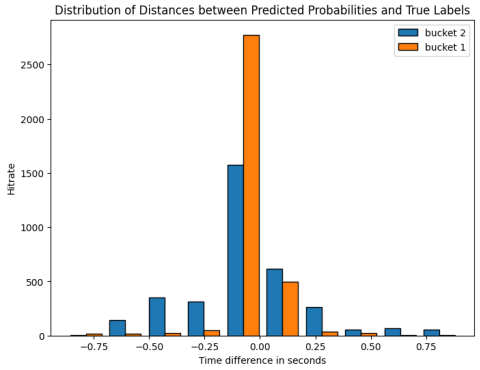

Fig. 8. GMP, Distribution of Distances between Predicted Probabilities and True Label

were the same, which makes us think that it is sensible to the initialisation and local minima.

IV. Discussion

Our models showed some encouraging results. On the MAPS dataset, the results were quite satisfying. On the GMP dataset, the results were not as good, but given the complexity of the dataset, it was to be expected. To improve the model further, it would be interesting to train it on a bigger and more varied dataset. Indeed, our training data was composed of several music pieces, but having more pieces, and thus more variety, would ensure that the model trains throu

V. Conclusion and Acknowledgment

We would like to thank Corvin Jaedicke and Fabian Seipel for their support throughout the term and Valentin Emiya for the provision of the MAPS dataset.

References

[1] R. Agrawal, S. Dixon, ”A Hybrid Approach to Audio-to-Score Alignment”, 2020
[2] C. Joder, S. Essid, G. Richard, ”Learning optimal features for polyphonic audio-to-score alignment”, IEEE Transactions on Audio, Speech, and Language Processing, 21(10):2118–2128, 2013.
[3] F. Henkel, S. Balke, M. Dorfer, G. Widmer, ”Score Following as a Multi-Modal Reinforcement Learning Problem”, 2019
[4] M. Dorfer, A. Arzt, G. Widmer, ”Towards score following in sheet music images”, 2016