ATIAM 2018 - ML Project
Multimodal embedding music for automatic piece recognition spaces

Mathieu Prang, Philippe Esling[*]
Institut de Recherche et Coordination Acoustique Musique (IRCAM)
UPMC - CNRS UMR 9912 - 1, Place Igor Stravinsky, F-75004 Paris
{mathieu.prang, esling}@ircam.fr

Abstract

This project aims to develop new representations for symbolic and audio music. You will try to implement the work done by Dorfer et al. [2018] and published in ISMIR 2018. Your goal is to represent musical symbols and the corresponding short excerpts of audio in the same space called *multimodal embedding space*. This approach allows to address the problem of matching musical audio directly to musical symbols. Moreover, theses kind of spaces could be very powerful tools for the orchestration field. By disentangling the correlation between the orchestral score and the audio signal result, we can provide efficient systems for analyze and generate specific orchestral effects. You will use Convolutional Neural Networks in order to capture features of the both modalities and represent them with vectors of the same dimension. First, you will have to prepare your dataset by synthesizing and aligning corresponding audio from MIDI files. Then, you will implement the model proposed by Dorfer et al which is composed by two networks and train it on your synthesized dataset. Once your model will be efficient on the training data, you will test it on real data through two tasks: (1) piece/score identification from audio queries and (2) retrieving relevant performances given a score as a search query. Finally, you will propose (or even implement) improvements in the architecture or the training of the model.

1 Introduction

In the past decades, the field of *computer music* has precisely targeted the problem of understanding musical concepts. Indeed, it is with this information that we can provide tools to help composers and listeners but also define methods of analysis and composition that improve our musical knowledge. Nowadays, a wide variety of approaches have been developed but on a large scale comparison, we can see that they all largely rely on one crucial point: *the way we represent music*.
In this project, you will use the machine learning framework to represent musical symbols and audio signal as points (vectors) in a common space. This kind of learning space could be very valuable for the musical analysis and composition field. First, this could lead to various analysis and knowledge inference tools. Furthermore, it could be an efficient new representation of music for many creative application.

1.1 Convolutional Neural Networks (CNN)

Convolutional NNs are an extension of classical NN that exploit the properties of *stationarity*. There are four main operations in this kind of network, each processed by a different layer (Figure 1).

[*] https://esling.github.io/

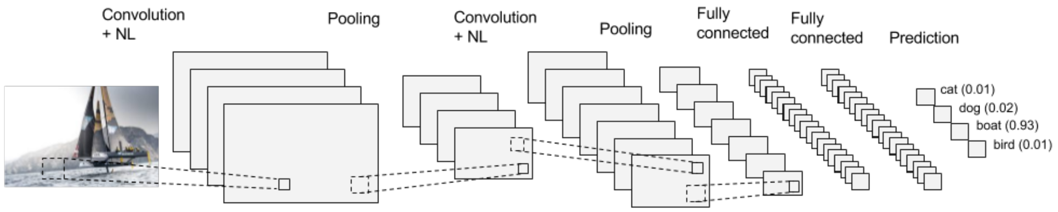

Figure 1: Convolutional neural network with two convolutional layers each followed by a pooling layer and two fully connected layers for classification.

Convolution The first layer is the convolution operator. Its primary purpose is to extract features from the input matrix. Indeed, each units k in this layer can be seen as a small filter determined by the weights W_k and bias b_k that we convolve across the width and height of the input data x. Hence, this layer will produce a 2-dimensional activation map h^k, that gives the activation of that filter across every spatial position

$$h^k_{ij} = (W^k * x)_{ij} + b_k \tag{1}$$

With the discrete convolution for a 2D signal defined as

$$f[m,n] * g[m,n] = \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} f[u,v]g[m-u, n-v] \tag{2}$$

The responses across different regions of space are called the *receptive fields*. During the training process, the network will learn filters that activate when they see some recurring features such as edges or other simple shapes. By stacking convolutional layers, the features in the upper layers can be considered as higher-level abstraction such as composed shapes.

Non-linearity We have to introduce *non-linearities* (NL) in the network in order to model complex relationships. Hence, before stacking every feature maps in order to obtain the output activations we apply a non-linear function like the recently proposed ReLU (Rectified Linear Unit) defined by $output = max(0, input)$ Dahl et al. [2013].

Pooling or subsampling Spatial pooling (also called subsampling or downsampling) allows to reduce the dimensionality of each feature map. The principle behind the pooling operation is to define a spatial neighborhood (such as a 3×3 window) and take the largest elements (*max-pooling*) or the average (*average-pooling*) of all elements in that window. In that way, we progressively reduce the size of the data representation to make it more manageable.

Classification Based on the highest level features in the network, we can use these to classify the input into various categories. One of the simplest way to do that is to add several fully-connected layers. By relying on this architecture, the CNN will take into account the combinations of features similarly to the multi-layer perceptron.

2 Dataset

Exercice 1 - Designing your training dataset Starting from one (or more) of the MIDI datasets listed bellow (of course, if you have your own MIDI dataset with Acid techno, Dub music or whatever you can use it but this could deteriorate the efficiency of the final model), you will construct a multimodal training set. To that end, you will follow the procedure describe in the section 2, 3.1 and 3.2 of the paper but with MIDI files (numeric scores) instead of sheet music for the symbolic part. For the symbolic representation of MIDI files you will rely on the piano-roll representation which is the traditional format largely used in music computing. However, the specificities of this piano-roll matrix makes it an ill-defined representation for learning. First, the resulting matrices are high dimensional (88 to 128 dimensions per time step) and the dynamics are usually encoded with a categorical distribution. Moreover, because of the typically small amount of notes played simultaneously, these vectors are usually highly sparse. This can be discussed in your last exercise where you will have to propose some improvements.
In the other hand, the audio part has to remain unchanged from the paper.

The following symbolic datasets which are made available for you can be consider as a reference in the symbolic music field and have been widely used in the literature.

- **JSB Chorales** is a corpus of 382 chorales by Jean-Sebastien Bach, which was splited into train, test and validation set by Allan and Williams [2005].
- **Nottingham** is a collection of 1200 British and American folk tunes [2].
- **Piano-midi.de** is a collection of classical music played on piano and splited by Poliner and Ellis [2006] [3].
- **MuseData** is a dataset developed by the CCARH that includes 880 orchestral pieces of famous composers [4].
- **LAKH dataset** is a collection of 176,581 MIDI files from various genres. Here, you can optionally use the subset of 17,256 files that have been cleaned to ensure the quality of the input data.

For the data augmentation part of the symbolic data, you will have to propose a new approach (or simply remove it, if it is justified) in order to fit with our MIDI scores.
Finally, once your dataset is ready, you will have to write a (very) short documentation on it with a small discussion about the potential learning differences between our MIDI approach and the sheet music one.

3 Models

All your implementations have to be performed by relying on the *Pytorch* framework for Python. Tutorials and documentation are available at `https://pytorch.org/docs/stable/index.html`

Exercise 2 - Implementing the model In this exercise, you will have to implement the model described in the section 3.3. Note that the first layers of the symbolic network may be slightly modified due to the change of the input representation.

Exercice 3 - Training with your synthesized dataset Here, you will refer to the section 3.3 of the paper in order to train your model with the loss and optimization procedure described by Dorfer et al. You will have to output the training graph with the loss values of both the training and test set and infer from it the best number of training epoch the model have to process.

4 Evaluations and improvements

Exercice 4 - Evaluation 1: Two-Way Snippet Retrieval By relying on the section 4 of the paper, you will evaluate the ability of your model to retrieve the correct counterpart when given an instance of the other modality as a search query. You can compare you results with those obtained by Dorfer et al. This evaluation has to be done on your synthesized data and then on real data. Do not forget to discuss your results.

Exercice 5 - Evaluation 2: Piece Identification and Performance Retrieval This second evaluation is described in the section 5 of the paper. Instead of trying to retrieve only snippets of the missing modality, you will extend the retrieval mechanism to complete pieces. This evaluation has to be done on your synthesized data and then on real data. Do not forget to discuss your results.

Exercice 6 - Improvements This question is open to any propositions that could produce interesting results. We will evaluate this on the justification and efficiency offered by the different improvements that you propose.

[2] https://ifdo.ca/ seymour/nottingham/nottingham.html

[3] http://www.piano-midi.de/

[4] http://www.musedata.org/

References

Matthias Dorfer, Jan Hajič Jr, Andreas Arzt, Harald Frostel, and Gerhard Widmer. Learning audio–sheet music correspondences for cross-modal retrieval and piece identification. *Transactions of the International Society for Music Information Retrieval*, 1(1), 2018.

George E Dahl, Tara N Sainath, and Geoffrey E Hinton. Improving deep neural networks for lvcsr using rectified linear units and dropout. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8609–8613. IEEE, 2013.

Moray Allan and Christopher Williams. Harmonising chorales by probabilistic inference. In *Advances in neural information processing systems*, pages 25–32, 2005.

Graham E Poliner and Daniel PW Ellis. A discriminative model for polyphonic piano transcription. *EURASIP Journal on Advances in Signal Processing*, 2007(1):048317, 2006.