Conditional GAN for Diatonic Harmonic Sequences Generation in a VR Context

Anna Shvets
FabLab by Inetum
157 Boulevard McDonald, 75019 Paris, France
anna.shvets@inetum.world

Samer Darkazanli
iMSA
Rue Clos Maury 82000 Montauban, France
darkazanli.samer@imsa.msa.fr

The use of AI models for music generation receives an important attention from scientific communities. Different architectures of deep learning neural networks have been applied for this specific task, such as Recurrent Neural Networks (RNN), Generative Adversarial Networks (GAN), Autoencoders, Variational Autoencoders (VAE) and Transformers. One of the important aspects of the generation process is the possibility to control the output by providing the input parameters, and a conditional generation was widely used in a computer vision domain to meet this need. In a present research we adopt the principles of conditional generation using GAN architecture and convolutions, applying them to a temporal domain, resulting in building a conditional GAN for diatonic harmonic sequences generation. The model is further used as a core feature of the VR module for computer-aided composition from “Graphs in harmony learning” VR project, where the sequence generation is conditioned by the user's input and the result is mapped to 3D representations of the generated chords.

Conditional GAN. Harmonic sequences generation. VR. Structural harmony method. Computer-aided composition.

1. INTRODUCTION

Recent advances in the field of deep neural networks influenced the approaches in computer-aided composition, which nowadays is enriched with a plethora of deep learning architectures completing a range of generation tasks. Those tasks correspond to the features of music language, such as generation of a melody (Yu *et al.* 2017; Yang, Chou & Yang 2017), accompaniment (Kosmas Kritsis *et al.* 2021; Dong *et al.* 2018, Kaliakatsos-Papakostas, Floros & Vrahatis 2012), temporal dependencies (Vohra, Goel & Sahoo 2014; Eck & Schmidhuber 2002) as well as the generation of entire sections of music in a specific style (Liang 2016; Huang *et al.* 2018). However, despite the variety of solutions, the conditional harmonic sequence generation in a symbolic domain, still leaves a room for experiments, and in a present paper we propose an adaptation of the Conditional GAN architecture, issued from a computer vision domain, preserving convolutional layers for computing sequential data. The idea of using such type of architecture was closely related to the use case of the model and is dependent from the VR application context, discussed further.

2. VR APPLICATION DESCRIPTION

The application “Graphs in harmony learning” aims to develop harmonic ear and the theoretical understanding of underlying relations of chords, attributed to function groups, via exploitation of the original representation of music knowledge based on the system of graphs. The understanding of music theory is aided with the use of a specific colour scheme, representing music functions, applied to the schematic representation of chords inside the system of graphs, as well as to the score representation of chords on the staff in a virtual 3D space. The explicit visual representation, achieved with the possibilities of the VR application context and proposed representation methodology, helps the learner to focus on a chord phonism and functional relationships between chords, which increases the overall learning efficiency (Shvets 2019).

A previous version of the application, presented before (Shvets & Darkazanli 2020), consisted of three modules – the explanation module (*Entry room*), explaining and illustrating the theory basics and a graph system to the learner, the main module (*Study room*), exposing the study material, and the testing module (*Test room*), challenging the

user with an auditory recognition of sequences, studied in the main module. A creative module (*Practice room*), is a new feature, which allows the user to apply in practice the acquired knowledge, using computer-aided sequence generation. This module has not been shown yet and will be further discussed in detail.

3. CONDITIONAL GAN

The AI model architecture choice is conditioned by the context in which the model is used, that is a creative module of the VR music learning application. As mentioned before, the module proposes a user the possibility to apply in practice the acquired knowledge about diatonic harmonic sequences, learned in a main module. Along with the test module, also based on an AI solution, the user receives a complete circle of learning experience, consisting of three phases – learning, testing and application in practice. This is the main reason why the existing models of music generation were not suitable, as the main restriction of this project consists in the obligation of generating diatonic harmonic sequences in a C key only.

Another requirement of the project was a participation of the learner in a generative process – the user should choose a chord from the system of graphs to trigger harmonic sequence generation, and the chosen chord will start the sequence returned by the model with a further mapping to its 3D staff notation representation. Therefore, the information about the chord should condition the content of the sequence to generate and be inputted as a class to the model. Giving these two main requirements, the architectural choice was made towards conditional GAN architecture.

3.1 Dataset and pre-processing

Before considering the architecture of the model, let us discuss the dataset and its preparation pipeline. In view of the diatonic content requirement, a classical along with romantic (early and middle periods) music styles were chosen to form a training set to the model, using the data present in Kaggle dataset *Classical Music*. The distribution of examples per composer as well as the number of extracted chords is shown in a table 1. The chord extraction itself was made via Python library for computational musicology *Music21* (Cuthbert & Ariza 2010). Additionally, manually constructed progressions, containing passing progressions, were added to the training corpus to ensure the presence of passing progressions in further generated sequences.

After the information extraction from MIDI files, the chord sequences were transposed to a C key and transformed to their roman numeral representation. A further filtering process consisted in the elimination of any chord alteration and sequential chord repetitions, since the generated sequences, limited to five chords only (due to a mapping 3D space limitations), should express a harmonic variety, free of repetitions. The prepared in such a way array of chords is then divided into seven classes, where the first chord is a representation of the class (one of seven degrees of the music scale), with the following four chords. Finally, the chords of each of sequences are replaced with their indices from the dictionary of unique chord values, converted to tensors and normalized between -1 and 1. In such a form the features are fed to the model along with their seven labels.

Table 1: The list of examples used for training.

Composer	Works	
	Number of MIDI files	Number of extracted chords
Haydn	21	193
Mozart	21	929
Beethoven	29	5355
Mendelsohn	15	1564
Shubert	29	10802
Chopin	48	5409
Liszt	16	7492
Brahms	10	2503
Total	189	34 247

3.2 GAN architecture

The network was built in Pytorch deep learning framework. The generator part of the model is composed of a tripled stack of transposed 1D convolution, batch normalization and rectified linear unit (ReLU) layers, with the addition of the final stack, composed of transposed 1D convolution and tangent layers. The discriminator part has one stack less, replacing transposed 1D convolution with a 1D convolution itself and utilizing leaky ReLU layer with a slope of 0.2, instead of ReLU, used in a generator. The final stack of the discriminator contains the convolution layer only (Figure 1). The Adam optimisation algorithm with a small learning rate (0.0002) was used for both parts of the model. Finally, binary cross entropy loss function (BCEWithLogitsLoss) was used to calculate the performance of both – generator and discriminator.

The information about the sequence label is added to the noise vector, being fed to the generator, while the label information for the discriminator is

Figure 1: Diagram of the network architecture.

added to the input channels dimension. This way both parts of the network have the information about the class to which the sequence belongs to: the generator for being able to transform the input noise tensor into the shape of the sequence, and the discriminator – to assess the effort of the generator, by comparing it to a ground truth example of the sequence inside a training loop.

3.3 Training

The extracted chords, being passed through the filtering process, resulted in a network input formation, containing 16 374 harmonic sequences - this data was used to train the generator and discriminator for 100 epochs. The learning curves for the generator and discriminator are shown in Figure 2.

Figure 2: The loss decrease during training for both models of the GAN.

The figure shows a good dynamic of the loss decrease for both parts of the network. The noisy shape of the curves is rather expected for GAN architecture. In the end of the training process, the two models were rather balanced with the generator loss equal to 0.74 and discriminator loss

– to 0.67, which are good metrics for GAN. The generation tests showed, however, that the model is prone to a mode collapse, therefore the network architecture may be further improved to achieve better stability in future works.

4. VR APPLICATION INTEGRATION

After the training process is finished, the best checkpoint of the model is exported to a binary file and integrated into the web application using Flask framework. The application containing the model is then deployed to the server and the communication with the VR application is done via JSON – the user of the VR application chooses the chord from a graph structure, which is sent to the web endpoint, where the model is deployed; the model takes that input and generates the rest of the sequence, sending it back to the VR application; upon the reception of the data, VR application adjusts the mapping of a sound and 3D models of chords and exposes a mapped version to the user as a sequence on the notation staff (Figure 3).

Figure 3: The sequence generated by conditional GAN, mapped to chords in a VR space.

5. RESULTS

In this paper we have shown an adaptation of the Conditional GAN architecture to sequential data, representing harmonic sequences of diatonic chords. We also demonstrated how this type of architecture, along with a generated content, serves to a specific use case, which is the integration to the creative module for computer-aided sequence generation in a VR harmony learning application. The perspectives of the work will focus on augmentation of the number of classes to include chord inversions into generation condition.

6. REFERENCES

- Cuthbert, M. and Ariza, C. (2010) "music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data," *International Conference on Music Information Retrieval*, pp. 637–642.
- Dong, H.W., Hsiao, W.Y., Yang, L.C. and Yang, Y.H., 2018, April. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. *AAAI Conference on Artificial Intelligence* (Vol. 32, No. 1).
- Eck, D. and Schmidhuber, J. (2002). Finding Temporal Structure in Music: Blues Improvisation with LSTM Recurrent Networks *12th IEEE Workshop on Neural Networks for Signal Processing*.
- Huang, C.Z.A., Vaswani, A., Uszkoreit, J., Shazeer, N., Simon, I., Hawthorne, C., Dai, A.M., Hoffman, M.D., Dinculescu, M. and Eck, D., 2018. Music transformer. *arXiv preprint arXiv:1809.04281*.
- Kritsis, K., Kylafi, T., Kaliakatsos-Papakostas, M., Pikrakis, A. and Katsouros, V. (2020). On the Adaptability of Recurrent Neural Networks for Real-Time Jazz Improvisation Accompaniment. *Frontiers in Artificial Intelligence*, 3, p.113.
- Liang, F. (2016). *Bachbot: Automatic Composition in the s Style of Bach Chorales*. M.Phil thesis. University of Cambridge.
- Shvets, A. (2019). Structural harmony method in the context of deep learning on example of music by Valentyn Sylvestrov and Philipp Glass. In: Weinel, J., Bowen, J.P., Diprose, G., and Lambert, N. (eds), *EVA London 2019 (Electronic Visualisation and the Arts)*, London, UK, 10–14 July 2019, 318–320. BCS, London. doi: [10.14236/ewic/EVA2019.60](https://doi.org/10.14236/ewic/EVA2019.60)
- Shvets, A., Darkazanli, S. (2020). Graphs in harmony learning: AI assisted VR application. In: Weinel, J., Bowen, J.P., Diprose, G., and Lambert, N. (eds), *EVA London 2020 (Electronic Visualisation and the Arts)*, London, UK, 6–9 July 2020, 104–105. BCS, London. doi: [10.14236/ewic/EVA2020.18](https://doi.org/10.14236/ewic/EVA2020.18)
- Vohra, R., Goel, K. and Sahoo, J. (2015). Modelling Temporal Dependencies in Data Using a DBN-LSTM. *IEEE International Conference on Data Science and Advanced Analytics*.
- Yang, L. C., Chou, S. Y., & Yang, Y. H. (2017). MidiNet: A convolutional generative adversarial network for symbolic-domain music generation. *arXiv preprint arXiv:1703.10847*.
- Yu, L., Zhang, W., Wang, J. and Yu, Y. (2017). Seqgan: Sequence generative adversarial nets with policy gradient. *AAAI conference on artificial intelligence* (Vol. 31, No. 1).