Cross Genre Music Generation using Image Representation

Abstract. We aim to solve the task of domain transfer for sequential data using convolutional neural networks. Given a set of unlabelled samples in two related domains, the objective is to learn a function that can map from source to target distributions. For this, we propose a novel image representation of sequential data and describe the use of convolution architecture to capture long term dependencies. Due to recent success of Generative Adversarial Networks (GANs) in modelling distributions, we employ a GAN architecture. In this respect, the Domain Transfer Network (DTN) of Taigman et al. present an approach for a related problem in image domain. In our work, we enhance the DTN using residual and skip connections to facilitate mapping of sequential data from source to target domain. Specifically, we apply our approach to one type of data (music) and demonstrate its ability to transfer music from one genre to another.

1 Introduction

Domain transfer is one of the most widely applicable problems which involves mapping samples from source distribution to target distribution. This can also be formulated in terms of making analogies between two related domains. There are several methods, which are inherently related but not identical, that tackle this problem to some extent. These include the use of GANs [1] to generate samples of a target distribution, style transfer methods [2] to alter the visual style of images, domain adaptation to generalize learned functions to new domains and transfer learning to import existing knowledge and to make learning much more efficient.

However, they do not completely address the general analogy synthesis problem. This can be formulated as : Given two domains S and T and a function f, the objective is to learn a mapping G such that $f(x) \sim f(G(x))$. Style transfer methods also solve a similar task but our problem is much more general in the sense that we consider an entire distribution as the target instead of one or few entities. The closest work related to our task is that of [3] where Tiagman et al. propose an approach for domain transfer in visual domain including digits and face images. But, to the best of our knowledge, there exists no prior literature for sequential data with the major challenge being modelling the temporal dependencies present in the data in relation to domain transfer task. We approach this task from the perspective of one type of sequential data, i.e., music using a novel image representation. We show how convolutional neural networks can be used in the context of sequential data, specifically describing the use of 2D spatial filters to capture temporal dependencies.

There are several other approaches, most notably Wavenet [4], which uses very deep dilated convolutional networks for modelling raw audio data. However, they focus on 1D dilated convolutions whereas we utilize 2D spatial convolutions (without dilation) and show how they be viewed as 1D dilated convolutions in temporal domain in the context of sequential data, which can be effective in capturing long term dependencies. The other additional advantage of using convolutional networks is speed improvement since we are not restricted to process the data in a sequential manner. This also enables the network to be trained on larger data in lesser time. Owing to these advantages, we use convolutional architecture in our task. Even though we work on music data, we emphasize that our approach is much more general in the sense that it can be applied to wide variety of sequential data.

The major objective of our task is cross-modal learning and adaptation between visual and audio domains and also to demonstrate the application of convolutional networks on sequential data in a novel manner. We aim to model musical data (in the form of MIDI files) using image representation and perform the task of cross genre music transfer. This is of significant interest not just from the theoretical point of view but can have wide practical applications also.

Our Contributions. We propose a novel image representation of sequential data to encode temporal features and describe the use of 2D spatial filters as 1D dilated convolution in temporal domain to capture long range dependencies present in the data. We also describe the modelling of short term and long term dependencies in deep layered structure of CNN architecture using the concept of increasing dilation. As an application novelty, we apply this method for the task of cross genre music transfer and demonstrate its ability to generate convincing music. Since it is very difficult to quantitatively estimate the quality of music generated, we evaluate our method in a qualitative sense by conduction a survey to analyze the quality of the generated music.

The paper is organized as follows: Section 2 gives a brief overview of related work and we give a precise formulation of our task in Section 3. Section 4 describes the novelty aspects and the architecture details of our methodology in detail. We begin by explaining the novel image representation for musical data in Section 4.1, demonstrate the use of 2D spatial filter as dilated temporal convolution in Section 4.2. We then present the architecture and loss function used for our task in Section 4.4 and Section 4.5 which are derived from [3]. The implementation details are mentioned in Section 5. Section 6 present the experimental results and discussion and finally, the paper is concluded in Section 7.

2 Related Work

Even though domain transfer for music has not been explored earlier, the problem of music generation has been extensively studied. So, we initially mention some of the related works in music generation and then move on to the use of CNNs.

The earlier techniques for music generation were based on graphical models which were manually defined by experts or learnt from examples [5, 6]. With the advancement of deep learning techniques, there are a number of endeavours where musical features such as notes, chords and notations are used to generate diverse music using LSTMs [7–10] to capture local note structure and learn long term dependencies. These involve structured input of music from MIDI files that are encoded as vectors and fed into an LSTM at each timestep. There are also approaches which make use of frequency spectrogram, the most recent being that of Donahue et al. [11] which uses GANs for synthesizing audio. Their SpecGAN and WaveGAN architectures show that GANs can be effective in modelling audio in both 1D and 2D space.

Other approaches exist which directly model raw audio samples, the best one being WaveNet [4] which uses a very deep dilated convolutional network. By increasing the amount of dilation at each depth, it is able to capture long range dependencies to obtain good representation of the audio. [12] investigate the use of CNNs in GAN framework to generate music and show comparable performance with Google's MelodyRNN. Inspired by the success of these works, we employ CNN architecture for cross genre music transfer.

Even though there is limited literature on domain transfer in music, extensive work has been done in the field of domain adaptation and domain transfer for images. Recent works like [13–16] clearly demonstrate that GAN and its variants serve as one of the best tools for the task of unsupervised domain adaptation. Architectures involving GANs have been shown to be effective in style/domain transfer tasks too as highlighted by [17, 18]. Specifically we would like to mention one work, that of Taigman et al., on which we base our work. The architecture described in their work, DTN, forms the core part of our architecture. They consider two major tasks - transferring digits from SVHN to MNIST and converting faces to emojis. Their results convincingly demonstrate that such an approach to domain transfer task can indeed be very effective.

We focus on the task of unsupervised genre transfer because of the lack of annotated musical data. Also, this task is challenging from the perspective of modelling sequential data like music using CNNs. With the recent success of GANs in modelling distributions, we use a GAN based architecture for unsupervised cross genre transfer. Our work is based on DTN which uses a compound loss function to transfer an image from one domain to another. We use MIDI audio files and convert them to MIDI images where the audio features at different time steps are distributed spatially across the image.

In terms of methodology, Bai et al. [19] also consider the use of convolutional networks for sequence modelling. They present an extensive empirical evaluation and comparison between recurrent networks and convolutional networks for multiple tasks. Their architecture, Temporal Convolutional Networks (TCN) is based on using dilated causal convolutions to capture long term dependencies inherent in sequential data. This is similar to our work but with the difference that we make use of simple convolutions (without dilation) and show that they are similar to dilated convolutions in temporal domain. Further, they consider

convolutions in 1D whereas we consider it in 2D using the image representation that we propose in this paper. Also, they consider causal tasks where the current time step depends only on previous time steps but the nature of our task and methodology allows us to capture information from both past and future efficiently. Based on the above mentioned several works and ours, we believe that convolutional networks can prove to be a powerful and effective task for tasks involving sequential data.

3 Problem Formulation

Given a set of unlabelled sequential data samples in a source domain S sampled $i.i.d$ according to some distribution D_S and a set of samples in the target domain T sampled $i.i.d$ from distribution D_T, we aim to learn a CNN based content extractor f, which captures long range dependencies in sequential data and a mapping $G : S \rightarrow T$ that minimizes a compound loss which consists of GAN loss, an f-constancy component (which requires that f is invariant under G), a regularizing component that encourages G to map samples from T to themselves and total variation loss. This is similar to [3] with the difference that we consider sequential data.

4 Methodology

4.1 Music space to Image space

Musical Instrument Digital Interface (MIDI) files are instructional files that explain how the sound should be produced once attached to a playback device or loaded into a particular software program that knows how to interpret the data. They explain the pitch and timestamp of notes played and are perfect for sharing musical information. This information can be represented as a MIDI image where each pixel (x, y) corresponds to timestamp x and pitch of y. This can further be extended to more than one feature by adding a third dimension z to encode multiple features like amplitude, MFCC coefficient etc. We propose a modified representation of MIDI images where each voxel (x, y, z) represents unique timestamp and value at this location represent pitch of this timestamp. Since, MIDI files consist of only pitch value, we propose a representation as shown in Figure 1. We use these modified MIDI images to provide as input to the CNN framework.

We use 3 channel modified MIDI image to give standard RGB image representation to music files thereby transforming music space to image space. This representation of sequential data in terms of images serves as a computationally effective method of processing such data as compared to RNNs for extracting features because neighbouring timestamps can be processed as a single convolution instead of sequential steps. Also, as we go deeper, the image size reduces as a result of which distant timestamps also come under the receptive field of filters. This becomes similar to dilated CNN structure of Wavenet with the only difference being that future timestamps are also considered in the filters.

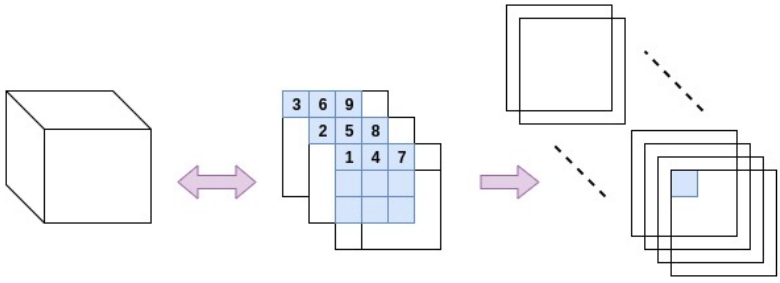

Figure 1. (left) MIDI image, (center) 3 channel representation with distribution of time instants where blue is 3x3 filter, (right) Feature map after one convolution where blue represents result of convolution

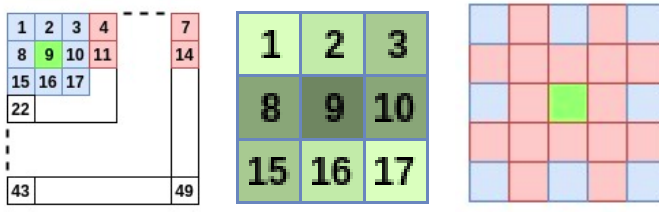

Figure 2. (left) 7x7x1 image, (center) 3X3 filter weight distribution (darker means higher weight), (right) 2D dilated convolution operation on 6X6 image (blue:pixels considered for convolution operation, green:center pixel, red:skipped pixels) Note: dimensions shown are for illustration purposes

4.2 2D spatial filter as 1D dilated temporal convolution

A dilated convolution is a convolution where the filter is applied over an area larger than its length by skipping input values with a certain step. This allows the network to operate on a coarser scale than with a normal convolution. Consider an unrolled version of the input MIDI image shown in Figure 3 and a 3x3 convolution filter shown in Figure 2 (left). Here, a 3x3 filter has a receptive field of 17 time instants including the future also but with intervals 4-7 and 11-14 skipped. This is similar to a 1D dilated convolution and enables the network to learn long term dependencies in a way similar to that done in WaveNet [4]. Further, since we use the filter weight distribution as shown in Figure 2 (center), the closer time instants are given more priority. As we move deeper into the network, the spatial dimension reduces leading to increased dilation and temporal receptive field. Hence, this can be considered similar to 2D spatial dilated convolution with gradually increasing dilation with the difference being that in our case, the increase is the inherent property. We argue that the deep CNN architecture in our scenario captures short term dependencies in the shallow layers and long term dependencies in the deep layers, owing to larger receptive field due to increased dilation.

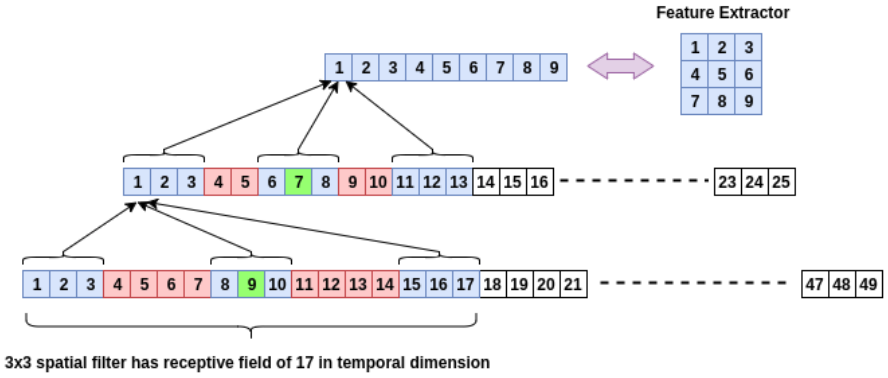

Figure 3. 1D dilated temporal convolution

4.3 Incorporating future information

The methods focused on the music generation operate by predicting the next time step conditioned on the past. This is done using RNNs or CNNs with masked filters. Since our task is focused on genre transfer rather than generation, we hypothesize that future information should also be included in the feature extraction process. This is also evident in other sequential data tasks where bidirectional RNNs are used but since we use CNNs, we leverage the future information using entire convolution filter instead of masking it. The weight distribution in filters is symmetric for past and future time steps.

4.4 Main Architecture

We employ the architecture proposed in [3] with minor modifications. The training data is unsupervised and consists of a set of samples from each domain. The modified Domain Transfer Network (DTN) used, employs a compound loss function that will be discussed further. The model is based on GAN framework. The generator is made up of encoder-decoder architecture which consists of convolutional layers where the encoder acts as feature extractor and the decoder as the image generator.The network structure is shown in Figure 4.

The major parts of the architecture are described below:

Content Extractor (f). This consists of an encoder to extract the desired features from the source image. It takes 32 x 32 x 3 MIDI images as input and generates feature vector.

Decoder (g). This consists of a decoder to generate the image in the target domain from the extracted features. It takes 1 x 1 x 128 feature vector as input and generates 32 x 32 x 3 image.

Generator is combination of content extractor (f) and decoder (g) and is represented as $G = g\ o\ f$

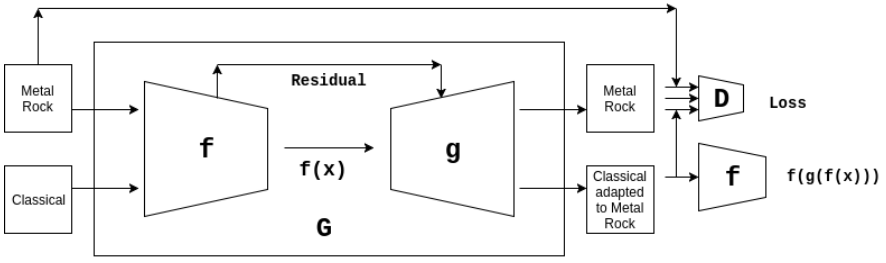

Figure 4. Network Architecture including residual connections

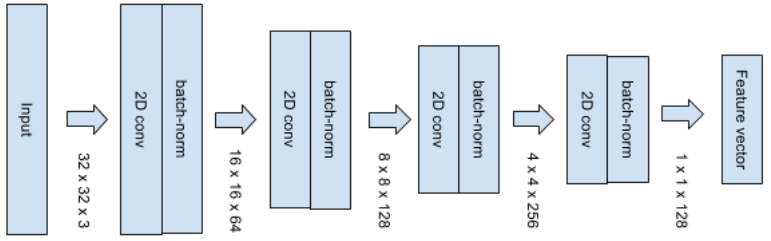

Figure 5. Encoder/Content Extractor Network

Residual and skip connections. We add residual and skip connections throughout the DTN architecture to speed up convergence. We are able to reduce the train time by significant amounts (3x times). In addition to improving convergence, the residual connections also enable transfer of long term and short term dependencies to be used by decoder while generating the sample in the target domain. This further reduces the f-constancy loss component since the decoder gets extra content information about the source domain.

Discriminator (D). This consist of a convolutional network where the task is to discriminate between the generated and the target images. It generates a probability value denoting how close the generator output is to the target domain.

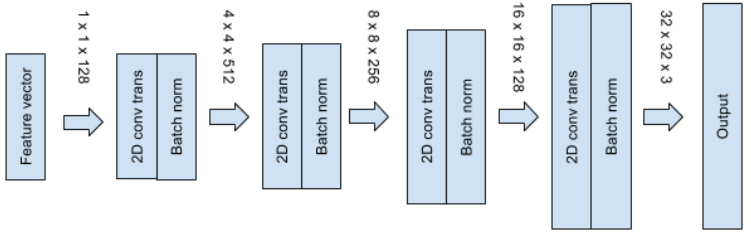

Figure 6. Decoder Network

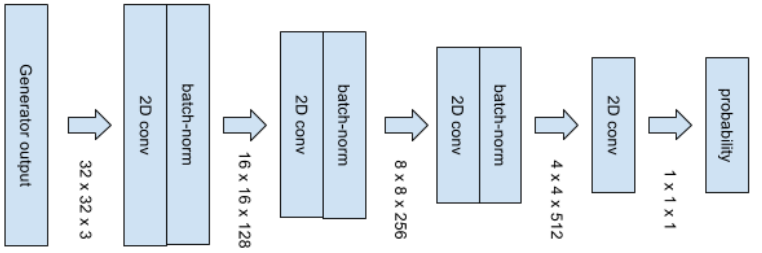

Figure 7. Discriminator Network

4.5 Loss Function

The mathematical formulation of each component in the compound loss function is provided below. The total generator loss is a weighted sum of L_{GANG}, L_{CONST}, L_{TID} and L_{TV}.

$$L_D = - \mathop{\mathbb{E}}_{x \in s} (log[1 - D(g(f(x)))]) - \mathop{\mathbb{E}}_{x \in t} (logD(x)) \tag{1}$$

$$L_{GANG} = - \mathop{\mathbb{E}}_{x \in s} (logD(g(f(x)))) - \mathop{\mathbb{E}}_{x \in t} (logD(g(f(x)))) \tag{2}$$

$$L_{CONST} = \sum_{x \in s} d(f(x), f(g(f(x)))) \tag{3}$$

$$L_{TID} = \sum_{x \in t} d_2(x, G(x)) \tag{4}$$

$$L_{TV} = \sum_{i,j} ((z_{i,j+1} - z_{i,j})^2 + (z_{i+1,j} - z_{i,j})^2)^{B/2} \tag{5}$$

$$L_G = L_{GANG} + \alpha L_{CONST} + \beta L_{TID} + \gamma L_{TV} \tag{6}$$

During optimization, L_G is minimized over G (generator as described above) and L_D is minimized over D. L_{CONST} represent the f constancy component, L_{TID} encourages generator to map samples from T to themselves and L_{TV} is the total variation loss component which is added in order to slightly smooth the resulting image. This is similar to the loss used in [3]. The generator and discriminator are trained in an alternating fashion from scratch using Adam optimizer.

5 Implementation Details

5.1 Dataset

We explored the Lakh MIDI dataset which consists of 176,581 unique MIDI files but due to the lack of annotations and many corrupt files, we use this

online resource[1]. It consists of 130,000 MIDI files having diverse genres like pop, classical, metal rock, EDM etc. We select the genres - classical as the source domain (S) and metal rock as the target domain (T) for our purpose. Classical has 10,245 files and Metal rock has 1748 files. These are converted to modified MIDI images and normalized in the range [-1,1] before feeding as input to our network. Consequently, the generated images are transformed to [0,255] and converted back to MIDI files.

5.2 Experimental Protocol

3 x 3 filters are used except at the last layer of the content extractor and first layer of the generator where it is 4 x 4. Input images are 32 x 32 x 3 and fed to the network in batches of size 128 each. ReLU activation is used except at the last layers of content extractor and generator where it is tanh since we need normalized output in the range [-1,1]. Adam optimizer is used to minimize the compound loss function. Learning rate is kept at 0.0003 and the weights are initialized using Xavier initialization. The code can be found here[2].

6 Results

To evaluate the aesthetic quality of the generated output, a user study that involves human listeners is needed. We conduct a survey with 50 participants. For each participant, a collection of 55 music files is given for him/her to classify into 4 categories :- blend, classical, metal rock and absurd. The 55 music files consist of 15 classical, 15 metal rock and 25 model generated outputs (ideally blend of classical and metal rock music). We classify each one of the 55 files into the 4 categories based on simple majority with threshold technique. Files getting a count of more than 20 for a class are classified as belonging to that class, else they are classified as absurd.

Table 1. Survey results for different categories of music files

Actual \voted	Blend	Classical	Metal Rock	Absurd
Blend	13	5	3	4
Classical	1	12	0	2
Metal Rock	1	1	11	2

Modified MIDI images of classical music files (Figure 9 (a) and (c)) don't have significant high frequency component. This plain structure is justified because classical music doesn't have sharp frequency changes. However, in the generated images (Figure 9 (b) and (d)), we can observe high frequency components owing to presence of metal rock properties infused in the classical image. Further, there

[1] https://www.reddit.com/r/WeAreTheMusicMakers/comments/3ajwe4/the_largest_mi di_collection_on_the_internet/

[2] https://github.com/Manthan-R-Sheth/domain-transfer-network

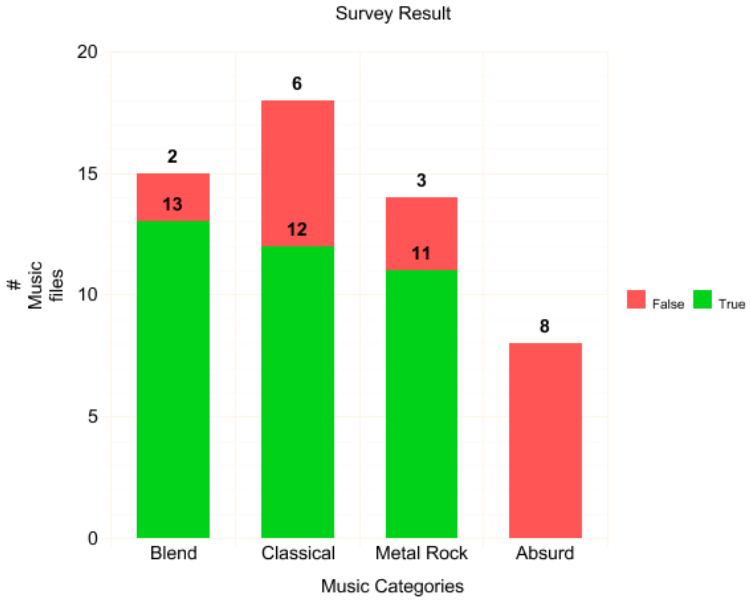

Figure 8. Precision value for each category. *True*(green) corresponds to the files belonging to correct class while *False*(red) corresponds to misclassified files.

is some inherent bias in the dataset we use, which leads to limited variations in the generated MIDI images. The genre of musical file can be accurately found if the file is large enough to encode all variations in music. A small portion of the file doesnt encode the necessary features to identify the genre of the musical file. Improving upon these limitations can further improve the performance of the model thereby giving better results.

7 Conclusion

We study the problem of domain transfer for sequential data using CNN architecture. We present a novel image representation for sequential data to encode

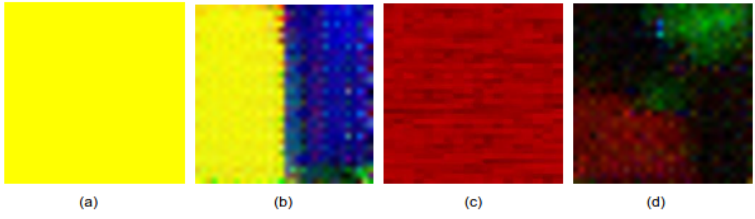

Figure 9. Results (a) and (c) input MIDI image of classical song. (b) and (d) corresponding output MIDI images of input files (classical + metal rock)

temporal features and describe the use of 2D spatial filters as 1D dilated convolution in temporal domain to capture long term dependencies. In addition, we explore the property of deep layered CNN architecture to model short term and long term dependencies and propose a novel application in terms of cross genre music transfer. The use of CNNs compared to RNNs result in efficient computation since the data is not processed at each time step. We enhance the encoder-decoder framework of DTN with residual and skip connections to generate convincing music. The initial results suggest promising prospects for modelling sequential data as images to apply convolutional networks. Also, our method is more general in the sense that it can be applied to wide variety of sequential data. As future direction of research, this approach can be applied to wide variety of sequential modelling tasks to test its effectiveness.

References

1. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2. NIPS'14, Cambridge, MA, USA, MIT Press (2014) 2672–2680
2. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (June 2016) 2414–2423
3. Taigman, Y., Polyak, A., Wolf, L.: Unsupervised cross-domain image generation. CoRR **abs/1611.02200** (2016)
4. van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A.W., Kavukcuoglu, K.: Wavenet: A generative model for raw audio. CoRR **abs/1609.03499** (2016)
5. Orton, R.: David cope, the algorithmic composer. madison, wi: A-r editions inc., 2000. 302 pp. with cd-rom (mac format). isbn 0-89579-454-3. Org. Sound **5**(2) (August 2000) 111–116
6. Nierhaus, G.: Algorithmic Composition: Paradigms of Automated Music Generation. 1st edn. Springer Publishing Company, Incorporated (2008)
7. Huang, A., Wu, R.: Deep learning for music. CoRR **abs/1606.04930** (2016)
8. Chen, C.C.J., Miikkulainen, R.: Creating melodies with evolving recurrent neural networks. In: Proceedings. IJCNN '01. International Joint Conference on Neural Networks. Volume 3. (2001) 2241–2246 vol.3
9. Liu, I., Ramakrishnan, B.: Bach in 2014: Music composition with recurrent neural network. CoRR **abs/1412.3191** (2014)
10. Eck, D., Schmidhuber, J.: A first look at music composition using lstm recurrent neural networks. Technical report (2002)
11. Donahue, C., McAuley, J., Puckette, M.: Synthesizing audio with generative adversarial networks. CoRR **abs/1802.04208** (2018)
12. Yang, L.C., Chou, S.Y., Yang, Y.H.: Midinet: A convolutional generative adversarial network for symbolic-domain music generation using 1d and 2d conditions. CoRR **abs/1703.10847** (2017)
13. Hong, W., Wang, Z., Yang, M., Yuan, J.: Conditional generative adversarial network for structured domain adaptation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (June 2018)

14. Hu, L., Kan, M., Shan, S., Chen, X.: Duplex generative adversarial network for unsupervised domain adaptation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (June 2018)
15. Zhang, W., Ouyang, W., Li, W., Xu, D.: Collaborative and adversarial network for unsupervised domain adaptation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (June 2018)
16. Volpi, R., Morerio, P., Savarese, S., Murino, V.: Adversarial feature augmentation for unsupervised domain adaptation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (June 2018)
17. Chang, H., Lu, J., Yu, F., Finkelstein, A.: Pairedcyclegan: Asymmetric style transfer for applying and removing makeup. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (June 2018)
18. Azadi, S., Fisher, M., Kim, V.G., Wang, Z., Shechtman, E., Darrell, T.: Multi-content gan for few-shot font style transfer. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (June 2018)
19. Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. CoRR **abs/1803.01271** (2018)