Symbolic Music Genre Transfer with CycleGAN

1. Project title: Symbolic Music Genre Transfer with CycleGAN

2. Short abstract of the project: Deep generative models such as Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) have recently been applied to style and domain transfer for images and in the case of VAEs, music. GAN-based models employing several generators and some form of cycle consistency loss have been among the most successful for domain transfer. In this project, we apply such a model to symbolic music and show the feasibility of our approach for music genre transfer. Evaluations will be done using separate genre classifiers created for the project. We closely follow the paper "Symbolic Music Genre Transfer with CycleGAN[3]". In order to improve the fidelity of the transformed music, we add additional discriminators that cause the generators to keep the structure of the original music mostly intact, while still achieving strong genre transfer. After replicating the paper[1], and it's results we will run the model on custom datasets. We plan to create a custom dataset, adding more genres to the existing 3 genres and then evaluating the model on this fresh dataset. Also, we plan to experiment with Virtual Adversarial Training[6] to train the discriminator.

3. Data: The original dataset[1] consists of 3 genres of music in the form of MIDI files.

- [Jazz - around 12k, Classical - around 16k, Pop - around 20k]

The MIDI files are pre-processed to form a matrix representation for each of 4 bar chunks. The sampling rate chosen for discretization of the time steps is 16 (as compared with the other literature) and notes above C8 and below C1 are ignored, resulting in an input matrix of shape 64 x 84. The 84 dimensions represent notes which are played simultaneously, with 1 indicating a note being played and 0 representing a note which is not played at that time step.
Newly Curated Dataset:

- [Rock, Classical, Jazz, Pop, Blues]

4. Techniques: We plan to implement CycleGANs from scratch for this task. At a time only two genres (A & B) would be trained with CycleGANs and therefore we will have 3 CycleGANs for ${}^3C_2$ pairs of training data (for 3 genres)[1]. For each CycleGAN we would have:

- 4 Discriminators[1]
    a. $D_a, D_b, D_{a,m}, D_{b,m}$
- 2 Generators[1]
    a. $G_{a->b}, G_{b->a}$

Please note that the number of CycleGans will increase to 10 for the ${}^5C_2$ pairs of training data in our new dataset.

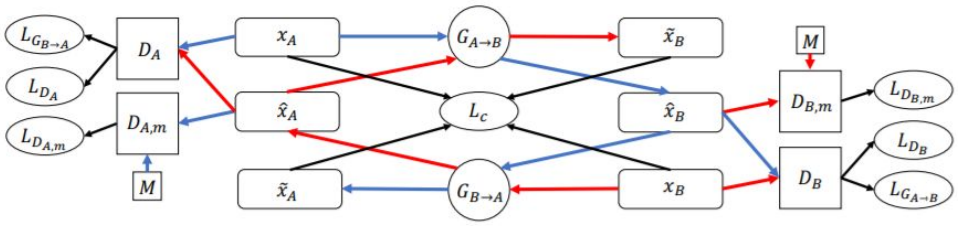

Fig. 1. Architecture of our model. The two cycles are shown in blue and red respectively. The black arrows point to the loss functions. We extend the basic CycleGAN architecture with additional discriminators $D_{A,m}$ and $D_{B,m}$.

- $G_{A\to B}$ and $G_{B\to A}$ are two generators which transfer data between A and B.
- D_A and D_B are two discriminators which distinguish if data is real or fake.
- $D_{A,m}$ and $D_{B,m}$ are two extra discriminators which force the generators to learn more high-level features.
- Following the blue arrows, x_A denotes a real data sample from source domain A.
- $\hat{x}_B$ denotes the same data sample after being transferred to target domain B, i.e., $\hat{x}_B = G_{A\to B}(x_A)$
- $\tilde{x}_A$ denotes the same data sample after being transferred back to the source domain A, i.e., $\tilde{x}_A = G_{B\to A}(G_{A\to B}(x_A))$. Equivalently, following the red arrows describes the opposite direction, i.e., the transfer from B to A and back to B.
- M is a dataset containing music from multiple domains, e.g, M = A ∪ B.
  x_M denotes a data sample from M. The intuition behind learning the discriminators, $D_{a,m}$, $D_{b,m}$ is to ensure that the style transferred sample produced from the generator are more "high level"
- Loss Function

$$L_{G_{A\to B}} = \|D_B(\hat{x}_B) - 1\|_2$$

$$L_{G_{B\to A}} = \|D_A(\hat{x}_A) - 1\|_2$$

$$L_G = L_{G_{A\to B}} + L_{G_{B\to A}} + \lambda L_c$$

$$L_{D_{A,m}} = \frac{1}{2}\left(\|D_{A,m}(x_M) - 1\|_2 + \|D_{A,m}(\hat{x}_A)\|_2\right)$$

$$L_{D_{B,m}} = \frac{1}{2}\left(\|D_{B,m}(x_M) - 1\|_2 + \|D_{B,m}(\hat{x}_B)\|_2\right)$$

where M denotes mixed real data from multiple domains (here possibly Jazz, Classic and/or Pop). Thus the total loss for the discriminators is

$$L_{D,all} = L_D + \gamma\left(L_{D_{A,m}} + L_{D_{B,m}}\right) \quad (2)$$

To enforce forward-backward consistency, Zhu et al. [2] introduce an extra L1 loss term called *cycle consistency loss*:

$$L_c = \|\tilde{x}_A - x_A\|_1 + \|\tilde{x}_B - x_B\|_1$$

$$L_{D_A} = \frac{1}{2}\left(\|D_A(x_A) - 1\|_2 + \|D_A(\hat{x}_A)\|_2\right)$$

$$L_{D_B} = \frac{1}{2}\left(\|D_B(x_B) - 1\|_2 + \|D_B(\hat{x}_B)\|_2\right)$$

All the Discriminators follow the same architecture and all the generators follow the same architecture as well.

The architecture of Discriminator [1]:

Input: $(batchsize \times 64 \times 84 \times 1)$					
layer	filter	stride	channel	instance norm	activation
conv	4×4	2×2	64	False	LReLu
conv	4×4	2×2	256	True	LReLu
conv	1×1	1×1	1	False	None
Output: $(batchsize \times 16 \times 21 \times 1)$					

Architecture of Generator [1] :

Input: $(batchsize \times 64 \times 84 \times 1)$					
layer	filter	stride	channel	instance norm	activation
conv	7×7	1×1	64	True	ReLu
conv	3×3	2×2	128	True	ReLu
conv	3×3	2×2	256	True	ReLu
$10\times$ ResNet	3×3	1×1	256	True	ReLu
	3×3	1×1	256	True	ReLu
deconv	3×3	2×2	128	True	ReLu
deconv	3×3	2×2	64	True	ReLu
deconv	7×7	1×1	1	False	Sigmoid
Output: $(batchsize \times 64 \times 84 \times 1)$					

5. Planned variations, improvements, alterations on existing works: The existing and the most recent work on a different architecture was using Seq2Seq [5], but as clearly mentioned in the blog, it does not perform as good as expected. However, here we are trying a completely different approach as researched by Brunner et al[1].

- Use Virtual Adversarial Training[6] to train the discriminators which will help us leverage unlabeled data samples to regularize the learning.
- Provide an open source implementation of the model in PyTorch, taking inspirations from [2][3]
- Curate a new dataset for this problem.
- Test the model for its successes and failures on the newly curated dataset and discuss the feasibility of this approach.

6. Input-output specifications
Input: Binary 2D matrix (64x84) piano roll representation for 4 bar chunks of music.
Output: Binary 2D matrix of piano roll representing the style transferred music.

7. Current Progress
We have finished the implementation of CycleGAN for this task https://github.com/shobhitd11/MUS206_CycleGAN_Music_Style_Transfer and would be working on replicating the results of the paper next in coherence with the proposed timelines and milestones. We would then take up newly curated datasets and applying Virtual Adversarial Training.

8. References

1. Symbolic Music Genre Transfer with CycleGAN, Brunner et al.
2. CycleGAN, https://hardikbansal.github.io/CycleGANBlog/
3. CycleGAN, https://junyanz.github.io/CycleGAN/
4. MIDINet, https://arxiv.org/abs/1703.10847.
5. Neural Translation of Musical Style, imanmalik.com/cs/2017/06/05/neural-style.html.
6. Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning https://arxiv.org/pdf/1704.03976.pdf