Group 13 Project Proposal

Jahya Burke*, Yisheng Ji†, Shiwei Rong‡, Shuangcheng Yang§ and Alan Yessenbayev¶

Department of Electrical and Computer Engineering
University of California, San Diego
La Jolla, USA

Email: {*j1burke, †y3ji, ‡srong, §shy003, ¶ayessenb}@ucsd.edu

I. Problem

For our project, we will address the problem of computer music composition. For thousands of years, cultures around the world composed music for recreational, religious, and artistic reasons. Up until now, music has been composed by humans but with the rise of machine learning there are opportunities for music to be composed using learning algorithms. Aside from the cultural benefits of music, there is now a large commercial industry surrounding music composition for entertainment venues, movies, video games, and retail. If a successful music composing application were to be created, it could be used for artistic as well as commercial benefits. Our system will be geared more towards EDM beat-makers and/or indie game developers, therefore we are aiming to have a simple interface that will produce simple musical ideas, that can be rehashed by human operator in real-time.

II. Dataset

In order to train our model, we will require large datasets of music. In particular, we decided to feed data in the MIDI file format, as it already contains the musical score information inside of it. The datasets will need to be restricted to a specific genre to avoid confusing the network with wide variation in compositional style. Based of examples that we have seen [1] [2] [3], we expect that different models will perform well for different genres of music. In order to allow for flexibility in the type of network we construct, we will consider composing 3 different genres of music; jazz music, which tends to be sporadic and free form, classical music, which tends to be dynamic and structured, and video game music, which tends to be formulaic and looping.

We found three datasets of interest. The Google MAESTRO [4] is a dataset containing over 172 hours of virtuoso piano performances. The LAKH dataset [5] is also a good dataset of MIDI files, however, it is not well-maintained and possibly contains corrupted files. Lastly, VGMusic.com [6] has more than 30000 MIDI files of almost all video game music out there. We will use a subset of this data to train our model.

III. Solution

We will explore auto-encoder neural network architectures to achieve our goal. A 96 notes with a time step of 96 ticks structure will be generated as the training dataset. This 96x96 sheet will comprise a single measure/bar. Additionally, a third dimension for the bars themselves may also be established. PCA and/or LDA methods will be used to compress the high-dimensional training data to a reasonable dimension for us to train our model. We will explore traditional, variational, and residual architectures. Time-permitting, we will extend the residual auto-encoder using the NODE method [7]. In the NODE method, instead of specifying a discrete sequence of hidden layers, the derivative of the hidden state is parametrized using a neural network and the adjoint sensitivity method is used to calculate the gradients. Ultimately, we will generate a composing AI interface. User can directly change a certain number of top principal components to change the style of music at any time. Also, notes in the piano roll format will be shown to the user. There may be other useful controls inside the interface to create various musical output.

ECE 271B Project, San Diego, US, 2018

IV. Experiments

We will use PyTorch [8] or TensorFlow [9] framework to train and test our model. To numerically assess the result of auto-encoder training, we will compute the test data reconstruction loss upon convergence for various architectures. However, primary goal of our system is to produce pleasing music, therefore the ultimate affirmation of the success of our model will be music that is pleasing for us. Also, we will assess our model, for testing speed, as we aim for the system to be used in real-time.

References

[1] CodeParade. (2018, Jul.) Generating songs with neural networks (neural composer). [Online]. Available: https://www.youtube.com/watch?v=UWxfnNXlVy8&t=3s

[2] carykh. (2017, Jul.) Ai evolves to compose 3 hours of jazz! [Online]. Available: https://www.youtube.com/watch?v=nA3YOFUCn4U

[3] ——. (2017, Mar.) Computer evolves to generate baroque music! [Online]. Available: https://www.youtube.com/watch?v=SacogDL_4JU

[4] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, "Enabling factorized piano music modeling and generation with the maestro dataset," 2018.

[5] C. Raffel. (2011) The lakh midi dataset v0.1. [Online]. Available: https://colinraffel.com/projects/lmd/

[6] M. Newman, S. Evans, M. Carroll, J. Hill, and D. Camarena. (2017) Video game music archive. [Online]. Available: https://www.vgmusic.com/

[7] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural ordinary differential equations," 2018. [Online]. Available: https://arxiv.org/abs/1806.07336

[8] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[9] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/