INFO 7390 Advances in Data Science and Architecture

Deep learning for Music Generation using Restricted Boltzmann Machine and TensorFlow

Under the guidance of

Prof. Nick Brown

Chetan M Jadhav | Manasa B Singhekar | Gaurav Sonkar

Jadhav.ch@husky.neu.edu, bhimarayasinghekar.m@husky.neu.edu , sonkar.g@husky.neu.edu

1.Abstract

The current focus of machine learning revolves around Neural Networks and it governs today's aspects of life. In the music industry, automatic music generation is the hot and trending research topic to explore the artificial intelligence in new domain. The purpose of the paper is to explore the effectiveness of the generated music and evaluation of the generated music for a genre. The combination of Restricted Boltzmann Machine and TensorFlow would provide insights on the model designed. This is a simple model with focus on the evaluation of music generated as there are complex algorithms to generate music but very few to explain the quality of music generated. The approach in this experiment is to generate music using python file that consists piano notes in the format of midi and the project uses midi file of "Pop" music genre.

2.Introduction

Music composition is an art, even the task of playing composed music takes considerable effort by humans. The complexity and abstractness, designing an algorithm to perform both tasks at once would be futile.
It is thus easier to model
this as a learning problem where composed music is used as training data to extract useful musical patterns.

The scope of the project remains restricted to one genre. Although, music generation for one genre is found undesirable, it's essential to focus on one genre to produce music that adheres to the music theory of that genre. Hence, we are using the aspects of music theory concepts such as notes and chords to aid the learning process. Our main contributions in this work are the incorporation of enforcing musical style in the model's output and learning music dynamics. The generated music at the end is audio signal in MIDI format. An obvious choice of architecture to model a time varying function would be a recurrent neural network because of its ability to share parameters across time. Specifically, we will be using Restricted Boltzmann Machine networks to model the signals.
We believe that the model we are designing can contribute the music composers and singers to produce their own creative music that is needed for movie or album. For example, a filmmaker may wish to match a video with music that is of a style to convey a desired emotion.

3. Related work

There has been a lot of work where musical features such as notes, chords and notations have been used to generate music using LSTMs [3,4,5]. These works show promising results and demonstrate that LSTMs can capture enough long-range information required for music generation.
A typical architecture in these approaches involves structured input of a music notation from MIDI files that are encoded as vectors and fed into an LSTM at each timestep.
The LSTM then predicts the encoding of the next timestep and this continues. The error is

computed as negative log loss of the predicted value and the true value.

There has also been work that directly models the raw audio samples, the best results are obtained from the paper [6] which uses a recurrent neural network and deep belief network to generate polyphonic music generation. The paper gives insights to a generic technique to model temporal dependencies and sequences using a combination of a recurrent neural network and a Deep Belief Network. The paper sheds light on using piano rolls to accommodate the training set with a defined piano notes and chords. The project involved using RNN-DBN with 2 hidden DBN layers - each having 150 binary units - and 150 binary units in the RNN layer. The visible layer has 88 binary units, corresponding to the full range of the piano from A0 to C8.

Recent work by Nayebi et. al [1] have also worked on audio samples, but instead of trying to learn and generate from raw audio samples, they work on the frequency domain of the audio. This approach is much faster because it allows the network to train and predict a group of samples that make up the frequency domain rather than one sample. Since the frequency domain can still represent all audible audio signals, there are no restrictions on the kind of music it can generate. They use a single LSTM architecture, where the samples in the Fourier domain are fed as input at each timestep. The LSTM generates the output which is the Fourier representation of the signal of the next timestep. The mean squared difference between the predicted output and the true frequency representation is used as the cost function to train the network. Because of good results and lower computation demands, we implemented this method as the baseline for all comparisons.

We found several drawbacks in the method, although there was a good tune present in the audio samples, it was covered with a lot of disturbance, the network failed to capture long range dependencies and sounded pleasant only in some regions lacking a coherent structure.

4.Approach

After consideration of each aspect for the project we decided to use combination of RNN- Restricted-Boltzmann-Machine and tensor flow to generate music from dataset of MIDI format. It consists of Pop genre with a total of 126 songs

.

4.1. Background

This section will provide details on the concepts and the reason for choosing amalgamation of RNN-RBM.

4.1.1 Dataset

The training data is around 126 MIDI files of pop songs.

MIDI (Musical Instrument Digital Interface)

MIDI (Musical Instrument Digital Interface) is a protocol designed for recording and playing back music on digital synthesizers that is supported by many makes of personal computer sound cards. Originally intended to control one keyboard from another, it was quickly adopted for the personal computer. Rather than representing musical sound directly, it transmits information about how

music is produced. The command set includes note-ons, note-offs, key velocity, pitch bend and other methods of controlling a synthesizer. The sound waves produced are those already stored in

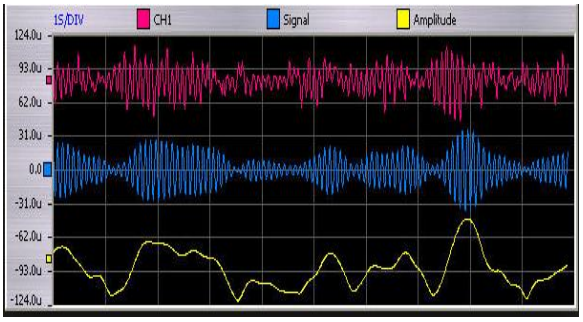
a wavetable in the receiving instrument or sound card.

Perhaps the best way to understand what MIDI is to first understand what it is not:

• MIDI isn't music

• MIDI doesn't contain any actual sounds

• MIDI isn't a digital music file format like MP3 or WAV



4.2 Technical Approach

Why Restricted-Boltzmann-Machine?

Our project needed the concept of generative models. Among the list of generative models in machine learning, we tested out dataset on each of them and below are the reasons for not choosing each one of them.

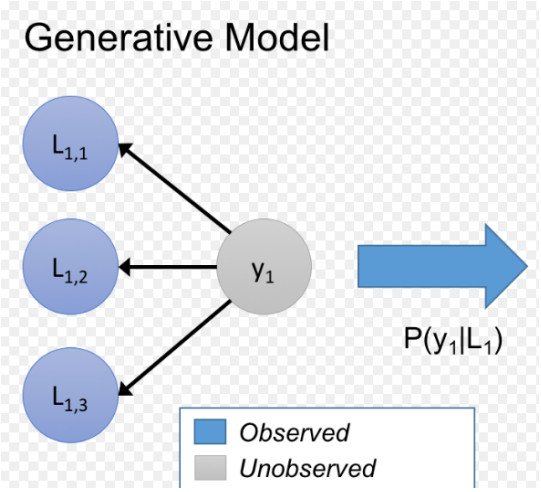
4.2.1 Generative models

Generative model is an approach for unsupervised learning. It is a model that allows us to learn simulator of data.

Generative models are typically probabilistic, specifying a joint probability distribution over observation and target (label)values. A conditional distribution can be formed from a generative model through Bayes' rule.

Generative Models specify a probability distribution over a dataset of input vectors. For an

unsupervised task, we form a model for P(x)P(x), where xx is an input vector. For a supervised task, we form a model for P(x|y)P(x|y), where yy is the label for xx. Like discriminative models, most generative models can be used for classification tasks. To perform classification with a generative model, we leverage the fact that if we know P(X|Y) P(X|Y) and P(Y)P(Y), we can use bayes rule to estimate P(Y|X) P(Y|X).



4.2.1.1 Restricted-Boltzmann-Machine

A restricted Boltzmann machine (RBM) is a generative stochastic artificial neural network that can learn a probability distribution over its set of inputs.

Architecture

The RBM is a neural network with 2 layers, the visible layer and the hidden layer. Each visible node is connected to each hidden

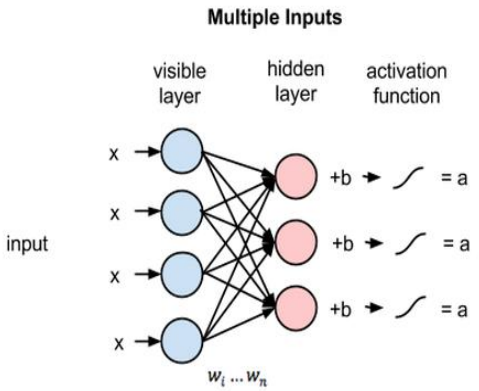
node (and vice versa), but there are no visible-visible or hidden-hidden connections (the RBM is a complete bipartite graph). Since there are only 2 layers, we can fully describe a trained RBM with 3 parameters:

• The weight matrix WW:

WW has size nvisible x nhiddennvisible x nhidden. WijWij is the weight of the connection between visible node ii and hidden node jj.

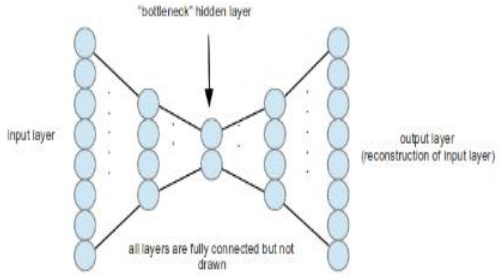
• The bias vector bvbv:

- bvbv is a vector with nvisiblenvisible elements. Element ii is the bias for the ithith visible node.
- The bias vector bh: bhbh is a vector with n_hidden elements. Element jj is the bias for the jthjth hidden node.
- nvisible nvisible is the number of features in the input vectors.Nhidden nhidden is the size of the hidden layer. Each visible node takes one chord. Each chord is multiplied by a weight and then the node's output at the hidden layer. Unlike most of the neural networks, RBMs are generative models that directly model the probability distribution of data.



Autoencoders:

An autoencoder is a type of artificial neural network used to learn efficient data coding in an unsupervised manner. The aim of an autoencoder is to learn a representation (encoding) for a set of data, typically for dimensionality reduction.

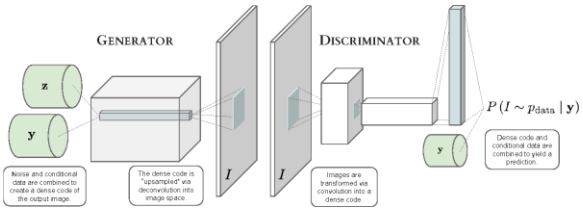


Although, the above neural network methods are used for image generation, these are the networks that could also be used for music generation. A combination of LSTM and Variational Autoencoders have been used to produce music [9].
Variational encoder for track generation involved that the first network (encoder) compresses the given track into a latent vector that works as a bottleneck. The second network (decoder) learns to reconstruct the melody out of a latent representation. This approach stimulates the network to work with a macrostructure of the track due to the low dimensionality of the latent vector.

In VAE, we optimize the lower variational bound

Generative Adversarial Networks:
Generative Adversarial Networks are models that allows us to learn simulator of data. Since, the data that we are dealing with is unlabeled data and falls under unsupervised learning. Generative models are mostly used in generation of text and photos.GAN has two differentiable functions, represented by neural networks, are locked in a game. The two elements (The generator and the

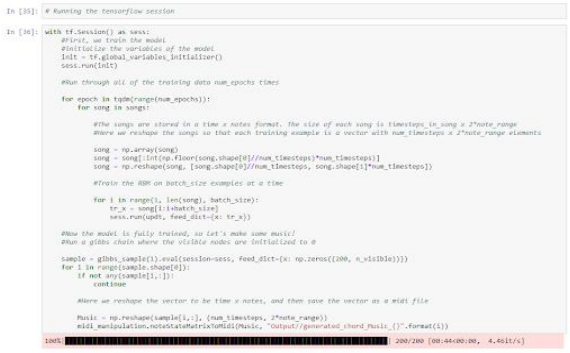
discriminator) have different roles in the framework. The generator tries to produce data that come from some probability distribution. The discriminator acts like the lie detector, as it gets to decide if the input comes from training set or generator. In the perfect equilibrium, the generator would capture the general training data distribution. As a result, the discriminator would be always unsure of whether its inputs are real or not.



GAN are best used for classification tasks, but since our project involves generating new samples from the training samples we first used GAN on our data set.
But, this model focuses mainly on optimization of loss function in the networks. Hence after evaluating each of the above listed models for out limited dataset and restriction we decided to choose Restricted-Boltzmann-Machine.

5.Code Implementation

After deciding the model, we used TensorFlow library to implement the model and a customized python file called midi manipulation that consists of piano notes



The generated midi files were then merged to form a single song of length 30s.

6.Discussion

6.1 Conclusion

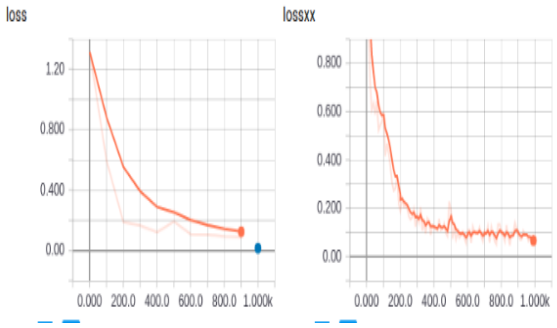
We could generate music that had chords of piano roll for the pop genre. To get a song of about 30s, the training time and the output file to generated it's about 2 minutes.

6.2 Evaluation

The evaluation of this model has two approach. Quantitative analysis and Qualitative analysis.

6.2.1 Quantitative analysis

The below tensor graph defines the loss in our model.



.

6.2.2 Qualitative analysis

The qualitative analysis involved conducting a listening experiment on 10 test subjects.

They rated our music on a scale of ten when given an option of identifying the genre. The result from this experiment wasn't satisfactory as the subjects mentioned that the chorus and verse is like metal songs as well.

6.2.3 Future work

We wish to develop a metric on music theory to evaluate the generated music for each genre.

7. References

1. Naseby, A., and Vitelli, M. 2015. GRUV : Algorithmic

Music Generation using Recurrent Neural Networks

2. Aaron van den Oord, Sander Dieleman, Heiga Zen,Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu

WaveNet: A Generative Model for Raw Audio

arXiv:1609.03499v2, 2016

3. Huang, A., and Wu, R, Deep Learning for Music:arXiv:1606.04930v1,2016

4. Daniel Johnson. Composing music with recurrent neural networks. *https : //goo.gl/Y P9QyR*

5. Google Magenta

https ://magenta.tensorflow.org/

6. Polyphonic Music Generation by Modeling Temporal Dependencies Using a RNN-DBN Kratarth Goel, Raunaq Vohra, and J.K. Sahoo https://link.springer.com/content/pdf/10.1007%2F978-3-319-11179-7_28.pdf

7. Sepp Hochreiter, Jurgen Schmidhuber,

Long short-term memory:

http://deeplearning.cs.cmu.edu/pdfs/Hochreiter7lstm.pdf

8.
https://mega.nz/#!Elq1TA7T!MXEZPzq9s9YObiUcMCoNQJmCbawZqzAkHzY4Ym6Gs_Q

9.Alexey Tikhonov1 and Ivan P. Yamshchikov2

Music Generation with Variational Recurrent Autoencoder Supported by History

8.Appendix

Github link:

https://github.com/Manasa9391/Music-Generation-with-RNN-RBM.git