Machine Learning Mandolin

by

Randy H. Nguyen

Frederik Meijer Honors College

Grand Valley State University

Honors Senior Project

HNR 499

Fall 2019

Faculty Advisor: Ira Woodring

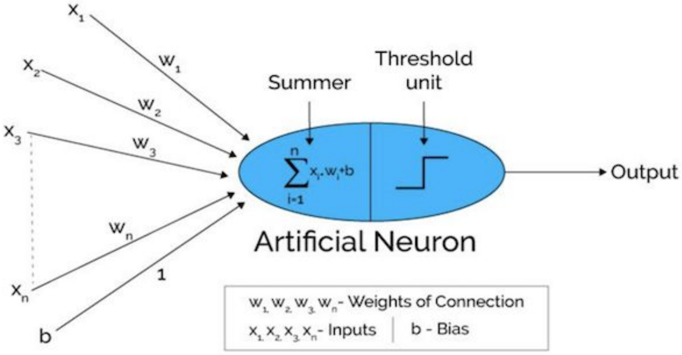
November 21, 2019

Part A: Introduction

Musicians combine their knowledge with intent to compose new musical pieces. Artists are endlessly creating more music, even though instruments have a limited set of sounds. Now, computer programs like Google's Magenta project use machine learning to assist musicians in creating new songs [1]. This is achieved by exposing a program to large amounts of music, and having the program learn patterns on its own rather than needing prior knowledge about music. The scope of this project will include researching how machine learning is applied to music generation and using a model to achieve this. Machine learning has been applied to music generation in many different styles, from classical piano, to jazz, and to video game music. This project's focus will be on mandolin music, specifically Vietnamese folk music. The objective is to input audio files into a program that after training on the data, will generate new music using the patterns that were detected.

Part B: Machine Learning Foundations

Machine learning is the application of statistics in computer models that make predictions based on a given dataset. Artificial neural networks (ANNs) were created to mimic how a biological brain would work. The smallest unit in an ANN is the perceptron, much like a neuron in a brain.



x_1
w_1
x_2
w_2
x_3
w_3
w_n
x_n
1
b
Summer
Threshold unit
$\sum_{i=1}^{n} x_i.w_i+b$
Output
Artificial Neuron
w_1 w_2 w_3 w_n - Weights of Connection
x_1 x_2 x_3 x_n - Inputs
b - Bias

Figure 1: Perceptron

The perceptron seen in Figure 1 sums the values of its inputs, and releases a signal depending on its activation function [2]. There are several activation functions, the one shown above is a step function where the output is 0 if the sum of the inputs is negative and 1 if the sum is positive. This particular activation function has discrete outputs, but there are functions with continuous values. Multiple perceptrons working together create a neural network, and a fully connected neural network is when each node from one layer is connected to every node of the next as seen in Figure 2.

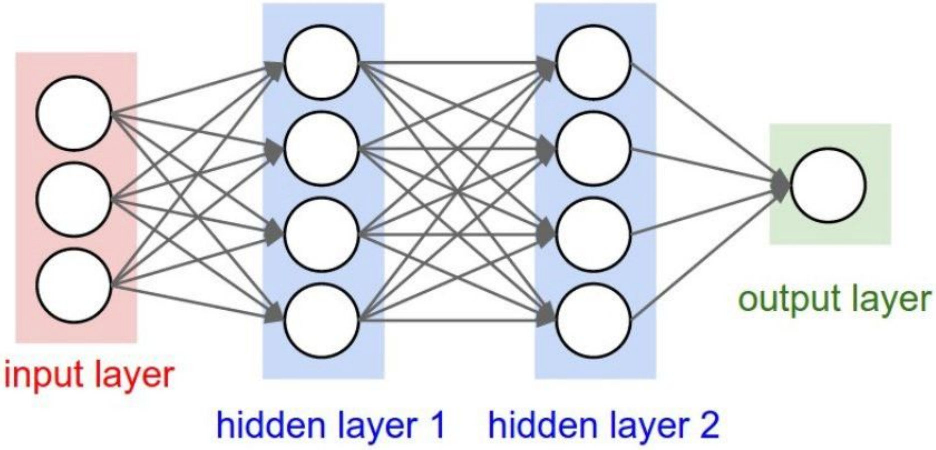


Figure 2: Dense Artificial Neural Network

Each of the connections have a weight associated with it. A perceptron can highly weight an input if is has a significant impact. As a neural network is trained, it is aware of the expected output and the actual output and loss functions score how well a model performed. The loss functions can be the absolute difference between expected and actual or the mean squared error among many other functions [3]. Neural networks will try to minimize the loss by adjusting the weights of the connections through a process called back propagation. ANN iterations involve the inputs being calculated and pushed forward by the perceptrons until the final output, which is used in a loss function with the expected result, and finally the weights are adjusted through back propagation. One enhancement beyond an ANN is the Long Short-Term Memory (LSTM) network. This type of network shown in Figure 3 allows a model to consider what occurred previously as well as the normal set of inputs [4]. The LSTM will play an important role in this project since musical notes have a logical relation with the sequence of preceding notes.

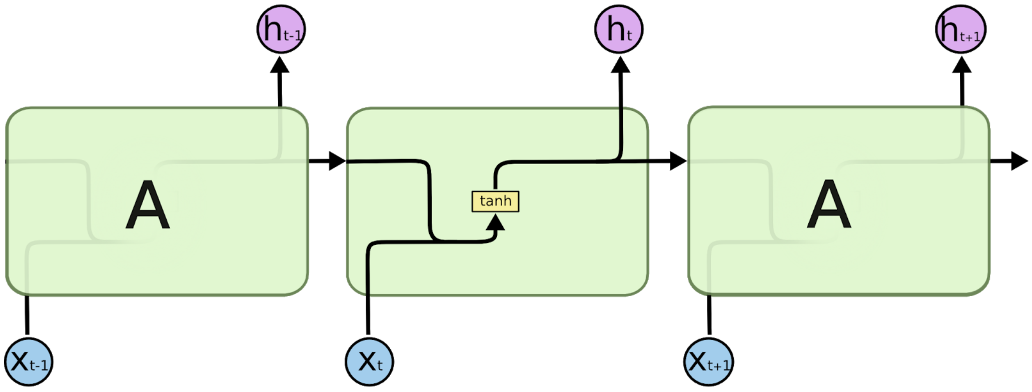


Figure 3: Long Short-Term Memory Network Diagram

Part C: Research

Initially, the plan was to use both Vietnamese mandolin music and other styles of music to build a model that could distinguish the difference between as the first step to music generation. The Fast Fourier Transform (FFT) for digital signal processing visualizes the most common frequencies in an audio file over time and a library is available in Python with an FFT function [5]. There may be different notes that are more common in different kinds of music. The intent was to teach the model to distinguish Vietnamese music from non-Vietnamese music as the first step towards generating music. However, the music files which were collected were very large, which caused this approach to fail.

Python is rising in popularity because of the library support that is available to everyone who wants to program with Python. Instead of creating custom functions, you could import a library that has been used by many others and trust that it performs the way you expect. One such library used while preparing the data in this project was LibROSA, which has functions for audio analysis. LibROSA's load function would import the audio file as a time series as well as provide the sample rate of the file [6]. This library was related to the FFT and FastAI, a library for classifying images, that used LibROSA to import files for the FFT [7]. However, there were models used in music generation, which meant that it was not necessary for classifying FFT images. LSTM models, as mentioned in the Machine Learning Foundations section, are a type of recurrent neural networks where previous calculations play a factor in the current calculation [8]. This is a logical choice for modeling music because a note depends on the sequence of notes preceding it. Therefore, the LSTM was the primary part of the experiments with different configurations of layers above it.

Machine learning involves numerous calculations that require significant computation power. Graphics cards (GPUs) are very efficient for machine learning because of their high core count, which allow them to perform multiple calculations simultaneously. NVIDIA is a GPU manufacturer that has a tool for optimizing GPUs for machine learning applications called the CUDA Toolkit [9]. This tool is necessary for GPUs to be discoverable by select Python libraries.

Part D: Data Collection and Preparation

Brian Oberlin, a mandolinist in the Grand Rapids area, permitted the use of his music in this project. He shared his YouTube channel, and pieces from his *Capriccio Fantastico* and *Troubadour* albums were mandolin solos and fitting for this project [10]. Several samples of Vietnamese mandolin songs were available on YouTube as well, although it was more difficult to find mandolin solos [11,12,13,14,15]. An online converter changed the video format into an mp3 file that only contained audio data [16]. Mp3 files are compressed to reduce the file size. To get an uncompressed file, another conversion tool was used that created wav files [17]. This format provides the most amount of data that could be utilized in a machine learning model.

Part E: Experiments

With the help of a starter program [8], different input parameters and model configurations were applied to try to generate new pieces of music. Two different computers were used to experiment with the data input as well as the LSTM model configuration. Primarily, a personal computer which utilizes an Intel i5-4960 CPU with 4 cores, 16 GB of DDR3 RAM, and an NVIDIA 960 GPU with 1024 cores and 2GB of GDDR5 was used. The university's research computer which contains an AMD Ryzen ThreadRipper CPU with 16 cores, 64GB of DDR4 RAM, and two graphics cards was also used. One is the NVIDIA Titan V with 5120 cores, 640 of which are TensorCores optimized for machine learning, and 12GB of GDDR5. The second card is the NVIDIA Quadro P6000 with 3840 cores and 24GB of GDDR5.

There were two issues that hindered the experimentation. First, there was an issue with X2Go, a remote desktop software, that caused the research computer to freeze and remove all sessions. This occurred before any longer training programs could finish. After the research computer shut down, a reconnection could not be established until the system administrator rebooted the computer. After diagnosing this issue with the system administrator, the likely source of the error appeared to be X2Go spamming the server with audio, file, and printer information from the client's computer. Experiments that had been planned for the research computer could not be accomplished on the personal computer because it did not have the available RAM for large lookback values or long iteration times. The second issue was with the personal computer where the GPU was not discoverable for the computations. This involved an HDF5 version mismatch that has not been able to resolve yet, but these experiments could still be conducted by suppressing this warning.

As these experiments are purely generating music and not classifying it, only samples of Vietnamese mandolin music were used to train the model. The initial result using the starter program's model was an audio file that nearly identical to the input, but the sound quality was worse because there was static. This initial experiment included the first 30 seconds of two songs that were concatenated into one data object where one song immediately followed the other.

The first hypothesis was to allow the model to learn from more data for a longer period. This was achieved by increasing the input data, the number of iterations, and the size of the lookback window for the LSTM. Keeping all other parameters constant, the number of input audio files increased to 12. Next, trials were conducted with 30, 50, and 100 training iterations to spend more time learning from the audio input. The 100 iteration trial could not be completed because there was a memory error during execution. The last parameter that was edited was the lookback value, which is the time window of previous signals that affect the current states. These values increased from 3 to 10, 100, 1,000, 2,000, and 5,000. The iterations on the 2,000 lookback value took 300s at first and gradually grew to 900s by the end of the training. This growth in duration was the most dramatic of all the conducted experiments. The first 13 iterations in the 5,000 lookback trial took an average of 750s until there was a memory error that stopped execution. The results from the completed experiments for this hypothesis were like the results with the original program which suggested that either the model needed to change, or the parameter values needed to be higher.

Reconfiguring the model had the highest likelihood of yielding results due to the hardware limitations of computing with a CPU. The model's input layer would stay as an LSTM, but the neural network layers following it used different activation functions and varied in number. The selection of activation functions used were linear, relu, leaky relu, and sigmoid. One notable experiment had five relu layers alternating with five leakyrelu layers, which resulted in the poorest audio quality. The different combinations of number of layers and the activation functions did not yield any results that were clearer than the original trial or new patterns of music. Using five sigmoid layers with the five leakyrelu layers did not result in any sound at all.

For a final experiment, the loss function of the neural network would change. Loss functions are used to correct a model's predictions as it is trying to minimize this value [18]. The starter program uses the mean squared error loss function, which is a common loss function in statistics. By squaring the errors, it is guaranteed that the values will be positive, but will also be very large. This allows the program to correct any predictions that were far from the actual value because the distance is magnified. However, when a logarithmic function is applied to the errors, the predictions that were far from the actual values would not be adjusted as drastically as before. After running a trial with the mean squared logarithmic error loss function, the audio quality is much worse than the original which was the expected result. This last experiment was evidence that although the overall results were not what had been planned, this project was a good learning opportunity that ended with an accurate prediction of this machine learning model's behavior.

Part F: Next Steps

The highest priority is to resolve the HDF5 version mismatch issue that is preventing the personal computer from using GPU for computations. Unlocking this extra computation power will increase calculation speeds and the amount of data that can be used per training iteration. Another priority is to change the model so that the LSTM is not the first layer. Convolutional neural networks (CNNs) are primarily used in image processing because of its ability to detect features. Using a CNN on the audio data to detect any key features and then connecting this layer to the LSTM may yield better results. The data is in an acceptable format, but machine learning applications perform better as the amount of data increases so collecting more songs will be helpful but not the main concern.

Resources:

1. Google. “Magenta.” *Google AI*, https://ai.google/research/teams/brain/magenta/.

2. Chris, Adi. “From Perceptron to Deep Neural Nets.” Medium, Becoming Human: Artificial Intelligence Magazine, 3 Jan. 2018, https://becominghuman.ai/from-perceptron-to-deep-neural-nets-504b8ff616e.

3. MOAWAD, Assaad. “Neural Networks and Backpropagation Explained in a Simple Way.” Medium, DataThings, 8 Oct. 2019, https://medium.com/datathings/neural-networks-and-backpropagation-explained-in-a-simple-way-f540a3611f5e.

4. “Understanding LSTM Networks.” Understanding LSTM Networks -- Colah's Blog, http://colah.github.io/posts/2015-08-Understanding-LSTMs/?source=post_page-----37e2f46f1714----------------------.

5. Mueller, John Paul, and Luca Massaron. “Performing a Fast Fourier Transform (FFT) on a Sound File.” Dummies, https://www.dummies.com/programming/python/performing-a-fast-fourier-transform-fft-on-a-sound-file/.

6. librosa development team. “LibROSA.” LibROSA, https://librosa.github.io/librosa/.

7. Hartquist, John. “Audio Classification Using FastAI and On-the-Fly Frequency Transforms.” Medium, Towards Data Science, 29 Nov. 2018, https://towardsdatascience.com/audio-classification-using-fastai-and-on-the-fly-frequency-transforms-4dbe1b540f89.

8. Sharma, Animesh. “Music Generation Using LSTMs in Keras.” Medium, Intel Student Ambassadors, 3 Jan. 2019, https://medium.com/intel-student-ambassadors/music-generation-using-lstms-in-keras-9ded32835a8f.

9. “CUDA Zone.” NVIDIA Developer, NVIDIA, 12 Sept. 2019, https://developer.nvidia.com/cuda-zone.

10. “Brian Oberlin - Topic.” YouTube, YouTube, https://www.youtube.com/channel/UCDJuFAU49Gnzeu302W9w1eg.

11. xekaraoke. “Xe Hoang.” YouTube, YouTube, https://www.youtube.com/user/xekaraoke.

12. “Trung Phạm Văn.” YouTube, YouTube, https://www.youtube.com/channel/UCI2K6NHcQ4v81XWKKFhJD2Q.

13. “BENBEN MUSICAL.” YouTube, YouTube, https://www.youtube.com/channel/UCP1hN9fhylwRuJGIHO1fWew.

14. “Linh Thanh Đàn Tranh.” YouTube, YouTube, https://www.youtube.com/channel/UCR_zK2tcmtu3WJdBdA5rToQ.

15. “Onionright.” YouTube, YouTube, https://www.youtube.com/user/onionright.

16. “YouTube to MP3 Converter - Convert YouTube to MP3 in Seconds.” OnlineVideoConverter.com, https://www.onlinevideoconverter.com/mp3-converter.

17. “Convert Audio to WAV.” Online, https://audio.online-convert.com/convert-to-wav.

18. Parmar, Ravindra. “Common Loss Functions in Machine Learning.” Medium, Towards Data Science, 2 Sept. 2018, https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23.