**CSC 4760 Final Project**

**Heartbeat Prediction Based On Music Characteristics**

**Abstract:**

Heartbeat changes when you listen to music. A simple neural network model consisting of two fully connected layers was built and fed characteristics of music as well as the change in heartbeat that resulted from the subject when the music was listened to. From this the model was reasonably able to estimate the change in heartbeat with a MSE loss of 33.

One paragraph summary of entire study

**Introduction:**

Heartbeat involuntarily changes based upon the audio that a subject listens to. This occurs because of the phenomenon known as “Sympatho-respiratory coupling”[[1]](#footnote-1). In general, this means that the heart of the listener naturally synchronizes from its current heart rate, to the beats per minute of the music listened to. If the heart rate of the listener can be predicted before they listen to the music, it means that music that changes their heart rate could be selected for them. This could have the impact of raising or lowering heart rate – which has been studied to have the impact of calming or exciting the subject[[2]](#footnote-2). This could be used for therapeutic such as calming down a subject to reduce strain on their heart, or for malicious applications such as calming down a subject while they drive so they fall asleep.

Music possesses other characteristics than beats per minute. This includes length of the music piece, pitch, change in tempo from the beginning to end, and time signature. Currently existing python libraries such as “Librosa” can gather these characteristics. The change in heartbeat based upon listening to music can be gathered as well by measuring the heartbeat before listening to a piece of music, and after. By comparing the characteristics to the resultant heartbeat, a neural network can hopefully begin making somewhat accurate predictions.

**Methods:**

**Training Data Gathering:**  
Heartbeat was measured using a fitbit, while music characteristics was gathered based upon the “Librosa” python library. This is demonstrated in the following code:  
A screenshot of a computer

Description automatically generated

Figure 1 – The Get Tempo and Get Length functions, which applied the Librosa library to a given mp3 file.   
A screenshot of a computer program

Description automatically generated

Figure 2 – The Get average pitch and Get Data functions which gathered all data that the Librosa library could produce. This was returned to a string that could be later printed into a .csv file that was read to create the training tensors.

After using this code in command line, it was found to be tedious, so Tkinter was used to build a widget to allow manually pointing to absolute file paths instead of handtyping it out, this allowed data to very quickly be entered and sped up the process considerably.

A screenshot of a computer

Description automatically generated  
Figure 3 – the Data Getter widget. This widget was used to speed up data gathering.

**Neural Network Model Attempts:**  
**First Attempts: Nearly No Training Data and Only One Fully Connected Layer:**

The first approach involved using a single fully connected layer and around 20 pieces of training data. The wishful thinking was that the strong relation between heartbeat and tempo would lead to quick results that would require little refinement. This was not the case, MSE loss for training data remained high at around 56, considering that this was squared error, this would mean that predictions would be off by as much as 7 beats per minute. This effectively amounts to guesswork and is not sufficient. However, I developed this initial version over 24 hours at a hackathon, and won “Best Interactive Media”[[3]](#footnote-3).  
  
A computer screen shot of a computer code

Description automatically generated

Figure 4 – The single fully connected layer – this did not return adequate results.

A screenshot of a computer

Description automatically generated

Figure 5 – MSE Loss for the single fully connected neuron layer. This high level of error is wholly inadequate.

**Second Attempt: 2 Fully Connected Layers**

I next attempted to use 2 fully connected layers. This reduced the loss, however upon inspecting the output tensors for the input values I discovered that the neural network had minimized loss by simply picking a value that was the average of all the possible outputs. In short, to minimize error between determining whether something was left or right, the neural network was always guessing “middle”.

A screenshot of a computer

Description automatically generated

Figure 5 – The fully connected layers reduced loss by simply guessing the average of all outputs, therefore achieving nothing. It should be noted that this average is close to my own resting heartrate (80bpm). Overfitting however means that the model is producing nothing of value.

At this point, I concluded that the model needed some help making the connection that I could clearly see. I reformatted my data to instead represent the change in heart rate, instead of the heart rate itself. This greatly improved accuracy.

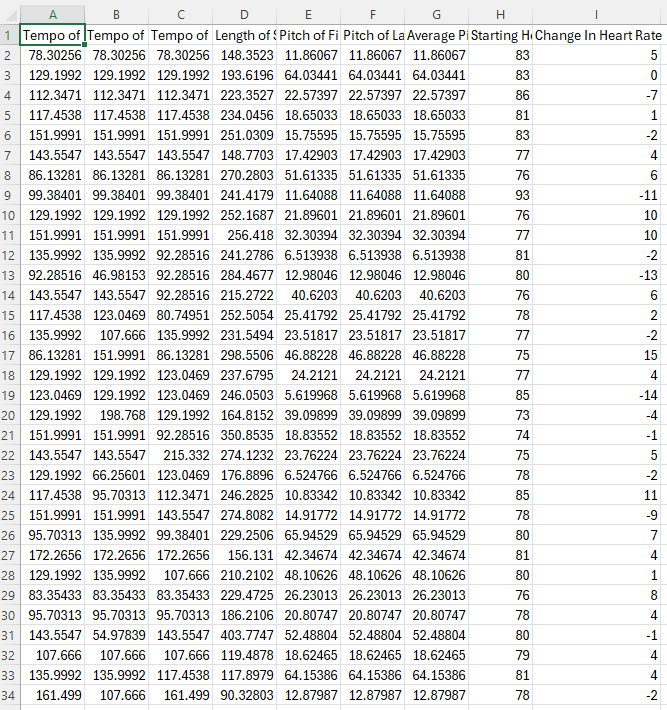


Figure 6 – Data reformatted to include change in heart rate.

A screenshot of a computer code

Description automatically generated

Figure 7 – Loss after adjusting the model to account for change in heart rate instead of just ending heart rate. Though the accuracy isn’t perfect, all cases where the heart rate change should be negative is negative (or near zero), and all instances where the heart rate change should be positive is predicted as positive. This means that the model is more functional as it can now reasonably predict the change in heart rate in one direction or the other.

**Discussion:**

The results are very promising, the fact that the direction of change in heartbeat can be somewhat reliably predicted on such few datapoints indicates that there is a strong correlation between music characteristics and change in heartbeat. Unfortunately, there is no quick way to gather this data – it involves a person listening to a piece of music – which inherently means the time to gather each piece of data is equal to the length of the music. Given that most songs average around 2 minutes long, this suggests that gathering even 100 samples would take at minimum 3.3 hours. However, with integration into a ‘walled garden’ type environment such as apple music on an apple watch, taking measurements on heartbeat at the beginning and ending of songs over the total population of apple watch users would be trivial – this suggests that the future scalability of the project is very feasible.

1. Watanabe, K., Ooishi, Y., & Kashino, M. (n.d.). Heart rate responses induced by acoustic tempo and its interaction with basal heart rate. PubMed Central (PMC). https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5339732/ [↑](#footnote-ref-1)
2. Trappe, H.-J. T., & Voit, G. (n.d.). The Cardiovascular Effect of Musical Genres: A Randomized Controlled Study on the Effect of Compositions by W. A. Mozart, J. Strauss, and ABBA. Dtsch Arztebl Int. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4906829/ [↑](#footnote-ref-2)
3. https://devpost.com/software/snappy-name-cooking [↑](#footnote-ref-3)