Joachim Alfonso A. Navarro

Lesli Natasha A. Coronel

May 2018

Real-time Identification of Common and Extended Musical Chords using

Artificial Neural Networks

Philippine Science High School

Main Campus

Submitted to the Faculty of the  
Philippine Science High School – Main Campus  
in partial fulfillment of the requirements for  
Science and Technology Research 2

May 2018

Real-time Identification of Common and

Extended Musical Chords using

Artificial Neural Networks

by

Joachim Alfonso A. Navarro

Lesli Natasha A. Coronel

ABSTRACT

Musical chords are fundamental to musical harmony and are named due to their importance. For most people, accurately identifying or naming musical chords is a difficult task requiring high levels of skill. With this in mind, a neural network that aimed to identify musical chord names from their component notes in real-time was programmed in Python using the Keras framework running on the TensorFlow library. Validation accuracy data was obtained after every training session, and manual MIDI inputs were used to obtain response time data. It was found that the current number of training iterations (2,400) provided an insufficient peak validation accuracy of 5.5%, but showed steadily decreasing values of the loss function. A left-tailed T-test for one mean was carried out on 30 randomly selected chords from the dataset and showed that the neural network responded significantly faster than the generally accepted standard of 10 milliseconds for real-time use. Further development of the neural network is recommended to increase validation accuracy. Such a neural network may be implemented on devices or software for the purposes of music education.

Keywords: Chord, Chord identification, Extended chord, GPU, Neural network, Real-time

Approval Sheet

This research work entitled, “**Real-time Identification of Common and Extended Musical Chords using Artificial Neural Networks**” by Joachim Alfonso A. Navarro and Lesli Natasha A. Coronel, presented to the Faculty of the Philippine Science High School – Main Campus in partial fulfillment of the requirements in Science and Technology Research 2, is hereby accepted.

|  |  |
| --- | --- |
|  |  |
|  | Kiel F. Granada  Research Adviser |

Acknowledgements

We would like to thank Ms. Kiel Granada, Mr. Edge Angeles, and Ms. Maria April Rose A. Andaca for their invaluable guidance and support, and Mr. Alejandro Alfonso Navarro and Ms. Maria Amina Amado for their financial assistance.

Table of Contents

|  |  |
| --- | --- |
|  | **Page** |

[Approval Sheet i](#_Toc412447683)

[Acknowledgements ii](#_Toc412447684)

[Table of Contents iii](#_Toc412447685)

[List of Tables v](#_Toc412447686)

[List of Figures vi](#_Toc412447687)

[Introduction 1](#_Toc412447688)

[Background of the Study 1](#_Toc412447689)

[Objectives of the Study 1](#_Toc412447690)

[Significance of the Study 1](#_Toc412447691)

[Scope and Limitations 1](#_Toc412447692)

[Literature Review 2](#_Toc412447693)

[Topic 1 2](#_Toc412447694)

[Sub-Topic 1.1 2](#_Toc412447695)

[Methodology 3](#_Toc412447696)

[Process Flowchart 3](#_Toc412447697)

[Process 1 3](#_Toc412447698)

[Process 2 3](#_Toc412447699)

[Results and Discussion 4](#_Toc412447700)

[Result of Process A 4](#_Toc412447701)

[Summary and Conclusion 5](#_Toc412447702)

[Recommendations 6](#_Toc412447703)

[Bibliography 7](#_Toc412447704)

[Appendix A’s Title Goes Here 8](#_Toc412447705)

[Sample Source Code 9](#_Toc412447706)

List of Tables

|  |  |  |
| --- | --- | --- |
| **Table** | **Title** | **Page** |
| 7 | Results of t-test for one mean on neural network response time | 17 |

List of Figures

|  |  |  |
| --- | --- | --- |
| **Figure** | **Title** | **Page** |
| 1  2  3  4  5  6  8 | Training (blue) and validation (orange) accuracies of neural network, epochs 0-800  Training (blue) and validation (orange) accuracies of neural network, epochs 800-1600  Training (blue) and validation (orange) accuracies of neural network, epochs 1600-2400  Training (blue) and validation (orange) losses of neural network, epochs 0-800  Training (blue) and validation (orange) losses of neural network, epochs 800-1600  Training (blue) and validation (orange) losses of neural network, epochs 1600-2400  Boxplot of total response times of neural network to MIDI inputs in milliseconds. Note: Null hypothesis is that response time is less than or equal to 10 ms. The program used to make this boxplot does not output units on the x-axis. | 13  14  14  15  16  16  17 |

Introduction

Background of the Study

Chords are collections of two or more musical notes, often played together, and are arranged in such a way that they follow the so-called “rules of harmony” (Leino, Brattico, Tervaniemi, & Vurst, 2007). These rules are recognized by humans as a response in the brain that is triggered when they are violated (Leino et al., 2007). Because of this, chords are fundamental to the harmonic integrity of any musical work.

A musical chord is commonly identified by three parameters: root note, chord type, and inversion. The root note serves as a reference point for the other notes which are played in the chord. These other notes are determined by the chord type. The inversion tells which note in the chord is played lowest. For example, a C major chord, 0th inversion has “C” as the root note, and “major” as the chord type. A major chord type includes the 1st (root), major 3rd, and 5th harmonics of the typical Western major scale, and the 0th inversion says that the root note is the bottom note. Thus, the notes of the 0th inversion of a C major chord are C (1st), E (major 3rd), and G (5th), in that order.

Humphrey, Bello, and Cho (n.d.) state that “the general music learning public places a high demand on chord-based representations of popular music” (par. 1). However, complete and accurate determination of these chords by hearing requires the use of both absolute and relative pitch, because chords utilize both an absolute reference point (root note) and a relative configuration of harmonies (chord type).

Absolute pitch is expressed when one can identify a musical note by hearing it, while relative pitch is shown when one can recognize the distances between musical notes (Zatorre, Perry, Beckett, Westbury, & Evans, 1998). While “most trained musicians” (Zatorre et al., 1998) exhibit a mastery of the latter, few of them have absolute pitch.

Absolute pitch is expressed in a low percentage of the human population and acquired through either favorable genes or music training at a young age, or both (Baharloo, Service, Risch, Gitschier, & Freimer, 2000).

Complete chord identification is thus a rare skill found in those with mastery of both absolute and relative pitch, even though chords play an important role in any musical work. An algorithm that automatically identifies chords from individual notes in real time would be a first step towards addressing this problem.

Artificial Neural Networks (ANNs) are computational models that use a layered structure of computational units called neurons in order to learn a certain task. Using mathematical functions, the neural network is able to train itself iteratively through a process called backpropagation until its error rate is significantly low (Nielsen, 2015; Sanderson, 2017). Thus, machine learning modeled through ANNs is designed to be adaptable and simple through self-organization (Daniel, 2013). The machine learns a given input dataset through by mapping an input through a probability distribution into its most probable class (Colina, Perez, & Paraan, 2017).

Objectives of the Study

This study aims to develop a neural network that quickly and correctly identifies common and extended one-root musical chords formed by playing more than two notes on a MIDI input device. Specifically, the program must identify common & extended chords and respond quickly enough to be used in live performance (Stark & Plumbley, 2009). The program must be implemented in programming languages that have MIDI input-output libraries such as pygame for Python and and neural network libraries such as Keras and TensorFlow to facilitate ease of coding.

Significance of the Study

Such application of real-time chord identification would be used in the field of music education, where a low proportion of music students have absolute pitch (Gregersen, Kowalsky, Kohn, & Marvin, 1999) despite their demand for chordal representations of music (Humphrey, Bello, & Cho, n.d., par. 1). These allow said students to learn to identify the chords they are playing more quickly and accurately, and help them develop their senses of relative and absolute pitch. They are also used in situations when musicians need to verify the chords they are playing for correctness, which usually happens when they are learning or composing a musical piece.

While automatic chord identification programs exist, they are either classical algorithm implementations that run on audio inputs in real-time (Fujishima, 1999; Stark & Plumbley, 2009), or neural network implementations that do not aim to run in real-time and do not include chords outside of major and minor triads (Perera & Kodithuwakku, 2005; Osmalskyj, Embrechts, Piérard, & Van Droogenbroeck, 2012; Zhou & Lerch, 2015). Including other chords such as 7ths and extended chords would allow identification of chords from more complex styles of music, such as jazz. The utilization of neural networks to identify extended chords from MIDI signals in real-time is largely unexplored and would provide useful data for future research.

Scope and Limitations

The study aims to create an artificial neural network that identifies both non-extended (“common”) and extended chords within a time limit of 10ms. The dataset comprises of the following chord types only: simple triads (major, minor, aug, dim), dominant extensions (7, 9, 11), major and minor extensions (M7, m7, M9, m9, M11, m11), suspended triads (sus2, sus4), major extensions with suspensions (M7sus2, M7sus4, M9sus2, M9sus4, M11sus2), dominant extensions with suspensions (7sus2, 7sus4, 9sus2, 9sus4, 11sus2), augmented and diminished extensions (aug7, dim7, ø7, aug9, dim9, aug11), and other extensions (mM7, mM9, M6, m6, M6(9), m6(9)).

MIDI commands will be used as inputs to the neural network. Audio datasets are thus not included in the scope of this study.

Literature Review

Chords

Chord structure and naming

Chords are the basis of harmony in music. According to the The Concise Oxford Dictionary of Music, a chord is “any simultaneous combination of notes, but usually of not fewer than 3” (Chord, 2004, p. 147). The name of a chord is dictated by two significant parameters: the root note and the chord type. The root note is a “note from which a chord originates” (Root, 2004, p. 615). It is taken as a reference point from which the chord type is determined. The chord type is also determined by the distances between the notes that comprise it.

Importance of chords amongst musicians

Humphrey, Bello, and Cho (n.d.) state that “the general music learning public places a high demand on chord-based representations of popular music” (par. 1). Websites on the Internet that provide chords for many popular songs exist, but these are generated by a user base of varying levels of skill and are not guaranteed to be correct or accurate.

Role of absolute pitch in chord identification

Identification of musical chords is dependent on absolute pitch, limiting the ability of the general population to identify chords. Absolute pitch is expressed when one can identify a certain musical note by hearing it (Zatorre, Perry, Beckett, Westbury, & Evans, 1998). Absolute pitch is naturally expressed in a low percentage of the human population and can be acquired through either a combination of favorable genes, music training at a young age (Baharloo, Service, Risch, Gitschier, & Freimer, 2000), or a comprehensive ear training course. Complete and accurate determination of these chords by hearing requires the use of absolute pitch, because chords are based on an absolute reference point, the root note (Root, 2004, p. 615).

Previous attempts at chord identification

Fujishima (1999) and Stark & Plumbley (2009) independently made classical algorithms for identifying musical chords from audio inputs in real-time. Both implementations were able to identify chords with a significantly high accuracy. Relatively successful implementations of chord identification using neural networks have also been attempted before, but they do not aim to run in real-time and do not include chords outside major and minor chords (Osmalskyj, Embrechts, Piérard, & Van Droogenbroeck, 2012; Perera & Kodithuwakku, 2005; Zhou & Lerch, 2015). Perera and Kodithuwakku (2015) further note that the use of neural networks to identify chords from MIDI signals is viable; however, none of these researches explore the use of neural networks to identify more complex or extended chords in real-time.

**Artificial Neural Networks (ANNs)**

Neural networks are computational models that use sets of computational units called neurons arranged together in a layered structure. Each neuron in one layer is connected to each neuron in the next layer. Each neuron has a number called its activation, and these activations are passed down the different layers of the neural network. When a neuron receives an activation from the previous layer, it takes on an activation by manipulating the activation passed on to it using a certain mathematical function. In essence, a neural network takes numbers as inputs and gives numbers as outputs (Nielsen, 2015; Sanderson, 2017).

Neural networks are trained to learn which set of outputs are expected for every set of inputs fed to it. This is done by a process called backpropagation. Training is performed for many cycles until the error rate of the neural network becomes significantly low and the accuracy becomes significantly high (Nielsen, 2015; Sanderson, 2017).

*Artificial Neural Networks in the fields of music*

Osmalskyj, Embrechts, Piérard, and Van Droogenbroeck (2012) proposed a feed-forward neural network for identification of recorded guitar chords and other instruments. A twelve-number vector called the Pitch Class Profile served as the input and represented each note in the chord, which the neural network would use to predict the chord. Another previous study suggested the use of deep neural networks to identify cover songs (Stamenovic, 2015). The method aimed to remedy the inability of other methods to study time and audio spectra by using the neural network to extract certain features of the input audio. Artificial Neural Networks have also been used to study patterns in Carnatic Raga classical music (Srimania & Parimalab, 2011).

**Hardware and software**

*Graphics processing unit*

Various pieces of hardware and software were utilized in this research. A GPU or graphics processing unit is a parallel processor (Nickolls, Buck, Garland, & Skadron, 2008) useful for parallel computing applications such as artificial neural network (ANN) simulations and training and is empirically found to be as much as 30 times faster than a regular CPU in these tasks (Colina, Perez & Paraan, 2017).

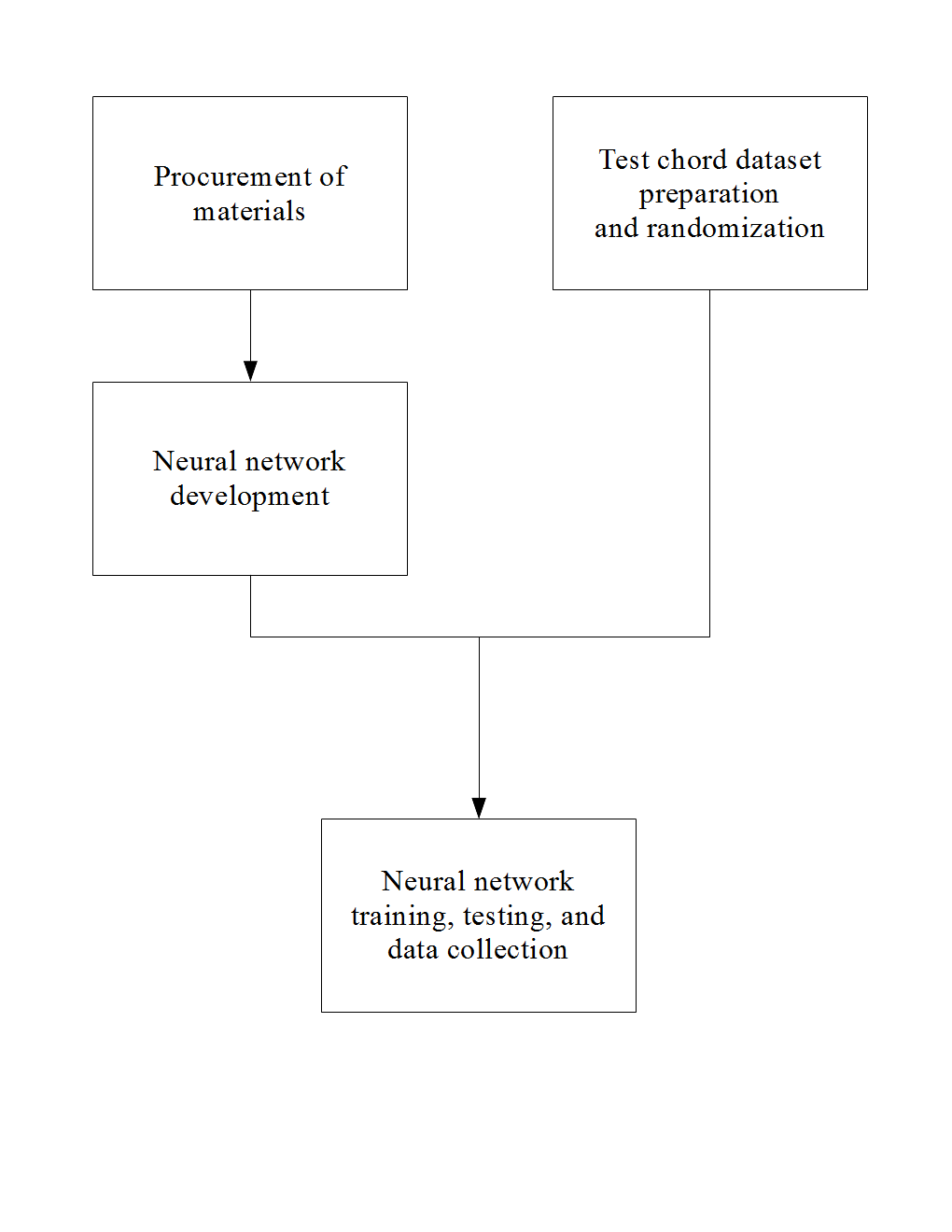
The NVIDIA GeForce GTX 1070 (“GTX 1070” or “1070”) is a GPU is used by many ANN researches (Colina, Perez, & Paraan, 2017; Kang, Hong, & Park, 2017; Kim, Hong, Nam, & Park, 2017, Zorrilla et al., 2017) and was thus selected for use in this research.

*Software*

Certain pieces of software will be necessary to define the structure of, train, and test the neural network. The neural network libraries necessary include Keras, a library that allows for the prototyping of the neural network on Python; as used by Xie, Zhang, Sapkota, and Yang (2016) in their research; and TensorFlow, as used by Covington, Adams, and Sargin (2016) which allows GPU parallelization (Colina, Perez, & Paraan, 2017) and serves as a backbone to Keras. Use of the GPU depends on NVIDIA CUDA which will be necessary for GPU computations and neural network simulation (Nickolls, Buck, Garland, & Skadron, 2008). Anaconda is a specific distribution of the Python and R programming languages specializing in scientific computing and data processing and will be used in programming the neural network. Finally, pygame (Shinners, 2017) will be installed as it contains facilities for real-time MIDI input. The pygame library is used in the successfully-implemented music platform developed by Cerny, Suthar, and Geselowitz (2013) for user-input MIDI and is therefore viable for use in research work.

Methodology

Process Flowchart



Procurement of materials

In order to facilitate computations that the neural network will perform, a graphics processing unit or GPU was utilized (Nickolls, Buck, Garland, & Skadron, 2008).

It is useful for parallel computing applications such as artificial neural network (ANN) simulations and training. Colina, Perez, and Paraan (2017) found that the use of a consumer GPU accelerated ANN computation by approximately 30 times when compared to a regular central processing unit. The minimum recommended GPU for research work is the NVIDIA GeForce GTX 1070 (“GTX 1070” or “1070”) (Dettmers, 2017). This GPU is used in many neural network researches and papers (Colina, Perez, & Paraan, 2017; Kang, Hong, & Park, 2017; Kim, Hong, Nam, & Park, 2017, Zorrilla et al., 2017) and was ordered online.

The necessary pieces of software to define the structure of, train, and test the neural network were acquired and installed in one of the researchers’ computers using the methods in mustgoplay (2016), which uses the following components.

Keras (Keras Team, 2018) is a Python-specific neural network library that allows for swift prototyping of ANN structures. Its relative simplicity is well-suited to research. It runs on TensorFlow, a neural networks library for Python.

NVIDIA CUDA & cuDNN are libraries that allow computation on an NVIDIA GPU and ANN simulation on a NVIDIA GPU, respectively.

Anaconda is a specific distribution of the Python and R programming languages for scientific computing and data processing.

PyCharm is a piece of software known as an independent development environment or IDE, which allows the user to write and run code.

Finally, pygame, a Python library for game development, will be installed as it contains provisions for real-time MIDI signal input to Python. This input can be fed to the neural network for chord identification.

Test chord dataset preparation and randomization

A program that makes a full list of chords from the 12 possible root notes and the chosen chord types was then written. The program outputs these chords to a text file that will serve as the input dataset of the ANN. Its training and validation datasets were randomized using a Python script to be detailed in Neural network training, testing, and data collection. The input dataset was split by this script into training and validation datasets with 80% and 20% of the chords respectively (Kolassa, 2015).

Neural network development

An algorithm that takes a current MIDI input and converts it into a format that can be interpreted by the neural network was written. The chords are interpreted by the neural network on a per-note basis (that is, one neuron per note spanning two octaves of 12 notes each for a total of 24 notes). Using the appropriate programming tools in Procurement of materials: Acquisition of programming tools, an artificial neural network whose goal is to identify chords was written in Python using Keras (Keras Team, 2018) and TensorFlow.

Neural network training, testing, and data collection

These processes are iterative by nature and thus proceeded once for every training epoch (one training “pass” through the dataset) of the neural network.

The neural network was trained using the built-in functions of Keras (Keras Team, 2018). During training, the neural network was tested with the chords in the training dataset output by the script in Test chord dataset preparation and randomization. The appropriate data such as training and validation accuracy and losses were recorded. Testing proceeded in an identical manner to training using the validation dataset. After the testing phase, adjustments were automatically made by Keras to the neural network depending on the desired output. The network was trained repeatedly until its validation accuracy stopped significantly increasing.

After training, the response time of the neural network was tested against a generally accepted standard of 10 ms for real-time use using a left-tailed T-test for one mean carried out on 30 randomly selected chords from the dataset. These chords were played manually on a MIDI controller, and response time was recorded using another Python script. The statistical T-test was carried out using R (R Core Team, 2017) and the Rcmdr package (Fox & Bouchet-Valat, 2017).

Results and Discussion

**Validation and training accuracies**

Every neural network is trained by traversing through a training dataset, and each pass through the training dataset is called an “epoch”. After every epoch, the accuracy of the network is tested separately using a different dataset called the validation dataset (Nielsen 2015).

After three training sessions of 800 epochs each, the neural network showed a markedly low peak validation accuracy, or rate of correct predictions on the test dataset, of under 5.5%. Every session of 800 epochs showed stepwise and irregular increases and decreases of validation accuracy before consistently dipping to 0% towards the end of the training session. Below are graphs showing the validation (orange) and training (blue) accuracies of the neural network.

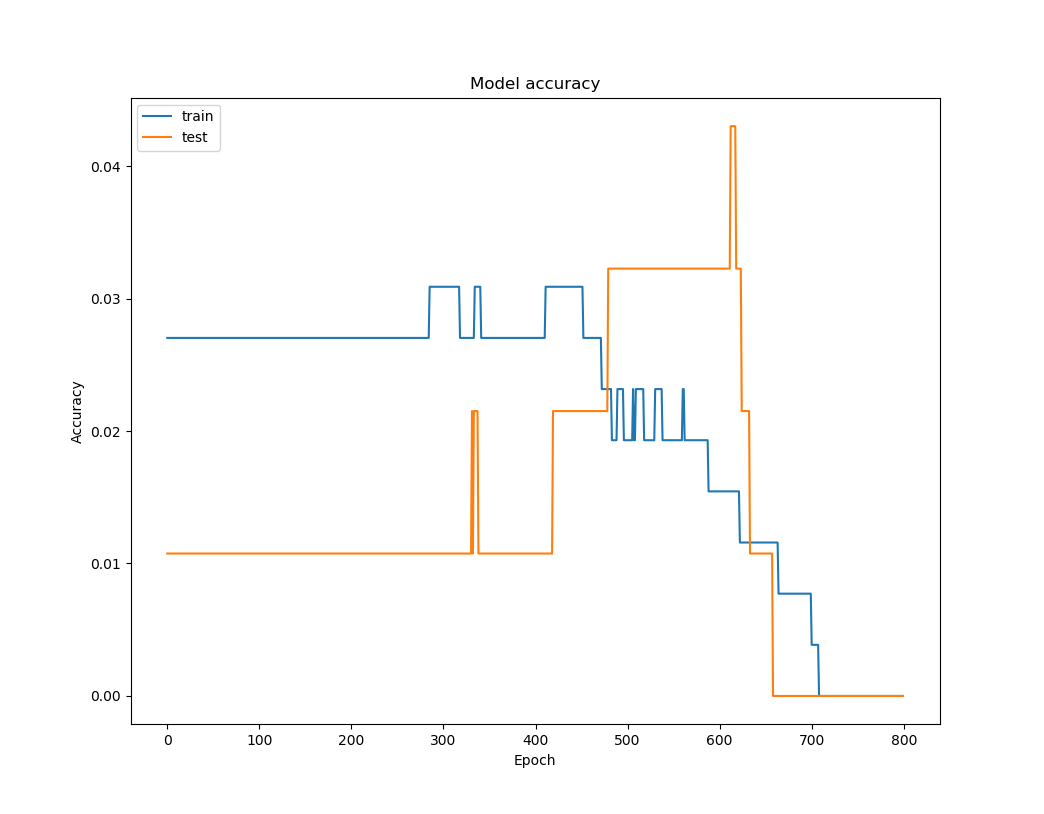


Figure 1: Training (blue) and validation (orange) accuracies of neural network, epochs 0-800.

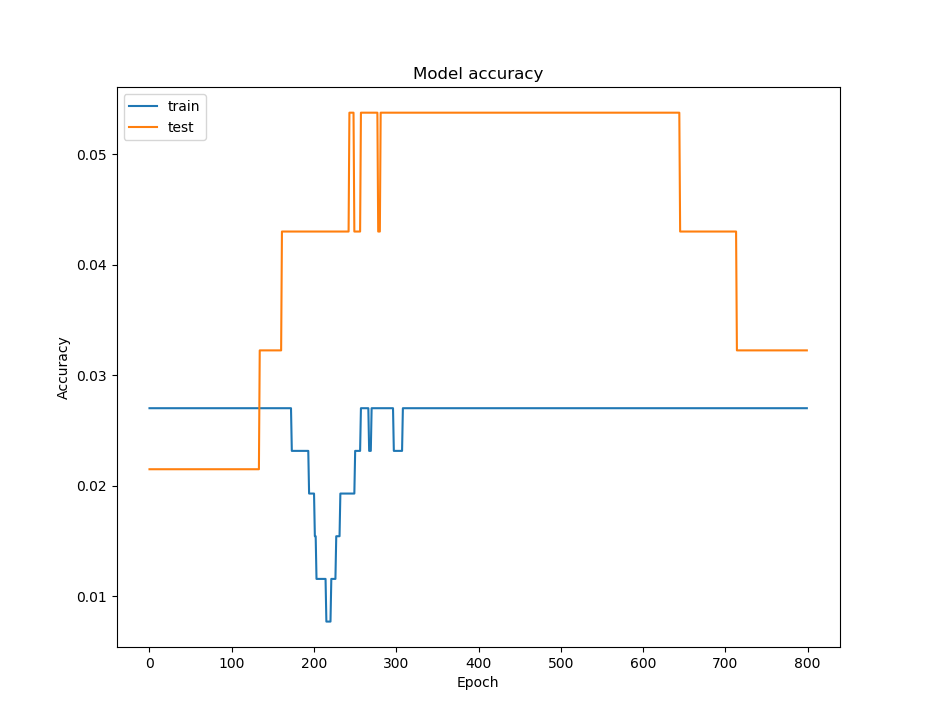


Figure 2: Training (blue) and validation (orange) accuracies of neural network, epochs 800-1600.

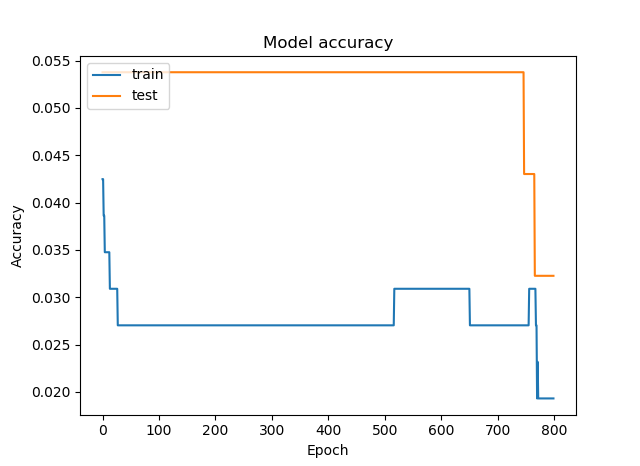


Figure 3: Training (blue) and validation (orange) accuracies of neural network, epochs 1600-2400.

Validation and training losses

The loss function is a special series of mathematical operators that shows how far the neural network is from the correct prediction. As such, lower values are considered favorable (Nielsen 2015).

The same training sessions returned decreasing values of the loss function as the training session proceeded. The training and validation losses consistently decreased in a smooth fashion as expected. Below are graphs showing the validation (orange) and training (blue) losses of the neural network.

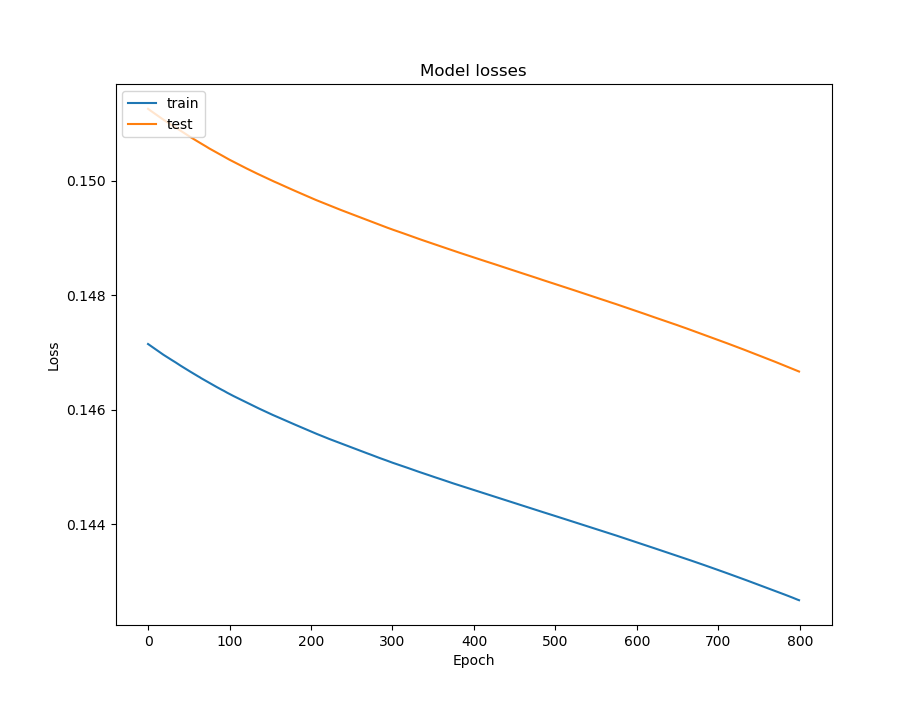


Figure 4: Training (blue) and validation (orange) losses of neural network, epochs 0-800.

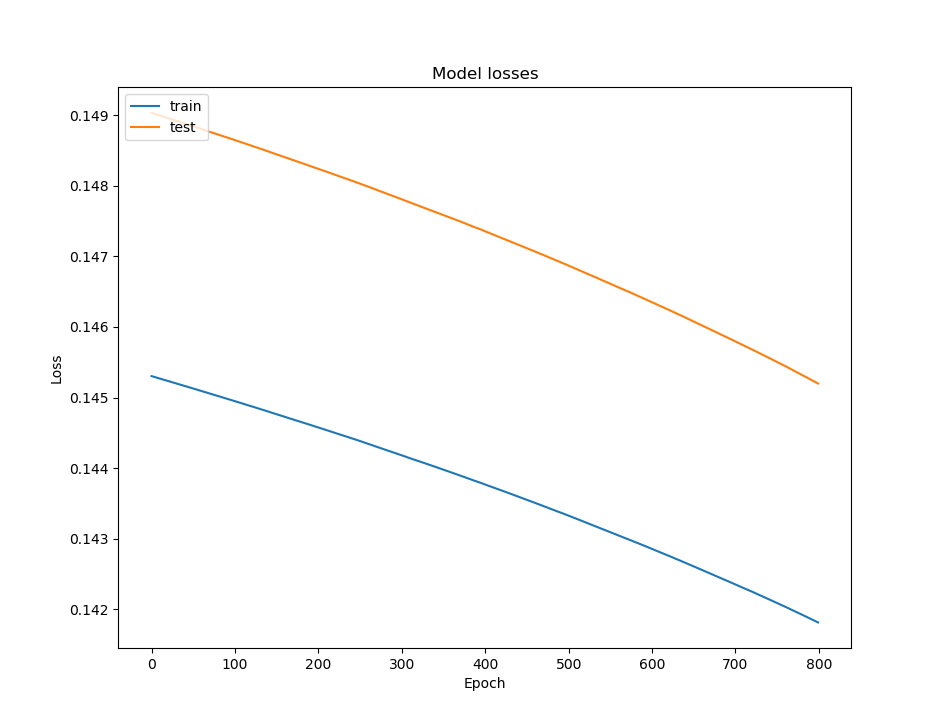


Figure 5: Training (blue) and validation (orange) losses of neural network, epochs 800-1600.

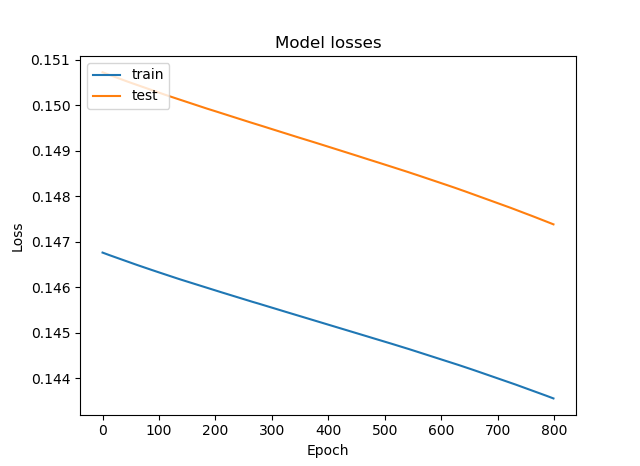


Figure 6: Training (blue) and validation (orange) losses of neural network, epochs 1600-2400.

Table 7: Results of t-test for one mean on neural network response time

|  |  |
| --- | --- |
| Summary measure | Value |
| Mean, ms | 2.95466667 |
| Mean t-statistic | -17.19 |
| Critical t | -1.699 |
| Null hypothesis outcome | Reject |

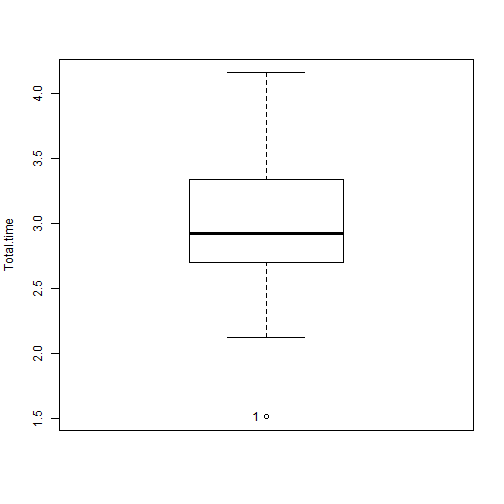


Figure 8: Boxplot of total response times of neural network to MIDI inputs in milliseconds. Note: Null hypothesis is that response time is less than or equal to 10 ms. The program used to make this boxplot does not output units on the x-axis.

The neural network performs atypically from the perspective of validation and training accuracies. It is expected that the training and validation accuracies would increase as the training proceeds. However, it is behaving as expected in terms of validation and training losses, as they decrease during training (Nielsen 2015).

Bodik (2018) suggests that this problem is caused by the input and output formats of the neural network being too complex for the neural network to properly classify input data. This can cause the neural network to underfit, i.e. not be able to adapt to the characteristics of the training data. This is manifested by consistently unfavorable values of validation accuracy. (Nielsen 2015, Bodik 2018). It is therefore recommended that steps to reduce this underfitting be taken in the future.

Summary and Conclusion

A neural network that aimed to identify common and extended musical chords in real-time was programmed, and did so with a markedly low accuracy. However, its satisfactory response time shows that neural networks may be used in real-time tasks.

Recommendations

The use of a simpler neural network structure or other machine learning algorithms like linear regression and an increase in training time are recommended.

Bibliography

Surname, G. M. (1985). *Title of work goes here: A subtitle*. City: Publisher.

Surname, G. M., & OtherSurname, F. N. (2011). Title of article. *Title of journal, volume(issue)*. Retrieved from http://someurl.goeshe.re/file.xyz

APPENDIX A

Appendix A’s Title Goes Here

Full data tables, letters of consent, or statistical test stuff go into one of a multitude of appendices.

APPENDIX B

Sample Source Code

If the project has program code, use the following style and format.

# Quicksort implemented in Python 2.x

# Always preface your source code with some comments describing the code

def Quicksort(Array):

# Check if we need to terminate recursion

if len(Array) <= 1:

return Array

else:

# Get pivot point from first element in Array

Pivot = Array.pop(0)

# Split Array into Left and Right arrays

Left = []

Right = []

for I in Array:

if I < Pivot:

Left.append(I)

else:

Right.append(I)

# Recurse, concatenate, and return the array

return Quicksort(Left) + [Pivot] + Quicksort(Right)