Joachim Alfonso A. Navarro

Lesli Natasha A. Coronel

May 2019

Real-time Identification of Simple 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  
Research 3

May 2019

Real-time Identification of Simple 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 by ear is a difficult task requiring high levels of skill and training. A neural network that aimed to identify musical chord names from their component notes in real-time was thus programmed. A dataset of 37 chord types with 12 notes each was used to train the neural network. Two sessions of training were conducted: one with 10,000 epochs and another with 20,000. Validation and training accuracy data were obtained after every session, and response time data was obtained by testing the network on manual MIDI inputs of chords randomly selected from the dataset. Peak accuracies of 0.2% and 7.5% were obtained on the validation and training datasets respectively. Overfitting on the training dataset was observed, suggesting that the task of choosing between 444 different chords is too complex for a neural network to perform. A left-tailed T-test for one mean (n = 30, α = 0.05) was carried out on response times on randomly selected chords from the dataset and showed that the neural network responded significantly more quickly (4.9 ms) than a standard of 40 ms for real-time use. The use of a different machine learning algorithm is recommended, as is using fewer chord types and other input formats such as audio. Chord identification algorithms may be implemented on devices or software for the purposes of music education.

Approval Sheet

This research work entitled, “**Real-time Identification of Simple 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 Research 3, is hereby accepted.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Mc Jervis S. Villaruel  Research 3 Teacher |  | Kiel F. Granada  Research Adviser |

Acknowledgements

Indicate your gratitude here. Keep it wholesome, and do not exceed one page.

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** |
| 1 | Label or Caption Goes Here | 1023 |

List of Figures

|  |  |  |
| --- | --- | --- |
| **Figure** | **Title** | **Page** |
| 1 | Label or Caption Goes Here | 1023 |

Introduction

Background of the Study

Musical chords are groups of individual notes often played together and are an integral part of harmony in many forms of music. They are arranged in such a way that they follow certain harmonic rules that are recognized by humans as a response in the brain that is triggered when they are violated (Leino, Brattico, Tervaniemi, & Vurst, 2007). Because of this, chords are fundamental to the harmonic integrity of any musical work.

A musical chord is commonly identified and named by its root note and type (hereinafter referred to as the “chord type”). 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. For example, a C major chord has “C” as the root note and “major” as the chord type. A major chord type includes the 1st (root), major 3rd, and 5th degrees of the typical Western major scale. Thus, the notes of a C major chord are C (1st degree), E major (3rd degree), and G (5th).

While “the general music-learning public places a high demand on chord-based representations of popular music” (Humphrey, Bello, & Cho n.d.), identifying the parameters of a certain chord takes certain skills obtained naturally or through special training (Zatorre, Perry, Beckett, Westbury, & Evans 1998), limiting the number of people who can correctly name chords by hearing them. 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 have absolute pitch.

Absolute pitch is expressed in a low percentage of the human population and acquired through either favorable genes, 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 in the context of music education.

Artificial Neural Networks (ANNs) are computational models that use a layered structure of computational units called neurons in order to learn a certain task. Neurons in one layer are connected to all neurons in the next layer; each link has a strength called a “weight”, and each neuron has a value called an “activation” and another value called a “bias”, which allows its activation to be manipulated. Using a calculus-based process called *stochastic gradient descent* (SGD), the neural network manipulates its weights and biases until a certain parameter is minimized or maximized. For example, a neural network can be configured to minimize the value of an error function, which indicates the inaccuracy of the network. (Colina, Perez, & Paraan, 2017; Daniel, 2013; Nielsen, 2015; Sanderson, 2017).

Objectives of the Study

This study aims to develop a neural network that quickly and correctly identifies one-root musical chords of simple and extended chord types formed by playing more than two notes on a MIDI input device. Specifically, the program must identify common & extended chords and respond within 40 milliseconds, a standard determined by Greeff (2016) for use in live performance. 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 implementation.

Significance of the Study

Current neural network implementations of chord identification such as that Perera and Kodithuwakku (2005) are limited to simpler chord types, and exploration of real-time chord identification using neural networks is limited to a few selected chord types: Perera and Kodithuwakku (2005), Osmalskyj, Embrechts, Pierard, and Van Droogenbroeck (2012), and Zhou and Lerch (2015) used two simple chord types only. Including extended chord types such as 7ths, 9ths, 11ths, and their permutations would allow the application of chord identification on more styles of music, such as jazz and blues. 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.

A possible application of real-time chord identification is a program that displays the chord name of a set of MIDI notes passed to it through a physical device such as a MIDI controller. Such a program would be useful 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.

Scope and Limitations

The artificial neural network was configured to identify both non-extended (“common”) and extended chords and aimed to display a result within a time limit of 40ms (Greeff, 2016). The following chord types were used: simple triads (major, minor, augmented, diminished), 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 notes were used as inputs to the neural network; audio datasets were thus not included in this study.

Literature Review

Topic 1

Discuss Topic 1 Here

To make it easier to generate the Table of Contents, whenever you add a new topic or sub-topic, highlight the topic heading and (in Microsoft Office 2010, 2013, and probably 2007) go to the **References** tab, and select **Add Text**. Choose **Level 2** (for topics) or **Level 3** (for sub-topics).

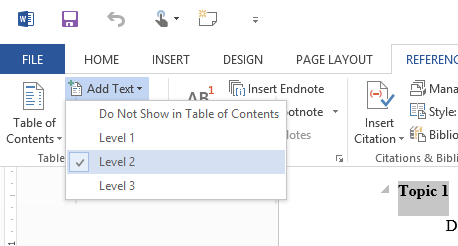


Figure 1: Setting a text entry to appear in the Table of Contents

Sub-Topic 1.1

Discuss Sub-Topic 1.1 Here, If Applicable

Methodology

Process Flowchart



Process 1

Discuss Process 1 Here

Any enumerations in bullet form should be single-line space, except in some instances. For example:

* A list should have at least five (5) items. Otherwise, use a straight sentence.
* If there are many lines per bullet, put in a blank line-space between entries.
* Third item to pad this example
* Fourth item
* Last item to make this worthwhile

Like so.

Process 2

Discuss Process 2 Here

Results and Discussion

Result of Process A

Table 1: Rater Preference of Pictures

|  |  |  |  |
| --- | --- | --- | --- |
| Rater | RATING OF PICTURES  (1 = awful, 5 = excellent) | | |
| Picture A  (Scenery) | Picture B  (Lolcat) | Picture C  (Fractal) |
| 1 |  |  |  |
| 2 |  |  |  |

Discuss Table 1 Here

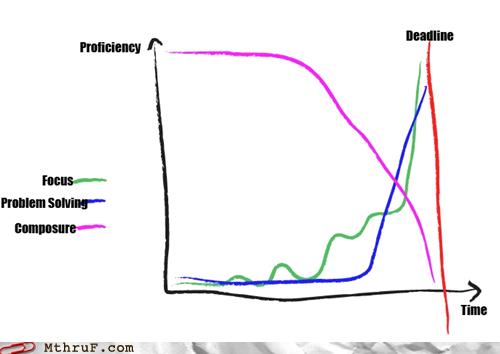


Figure 2: Graphs go here.

Summary and Conclusion

Summarize key results. Note the most important results and related to the objectives of the study.

Draw a conclusion based on the results. State if the project was successful or otherwise.

Recommendations

Recommendations Go Here

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)