Accademia Musicale Studio Musica

Michele Della Ventura, *editor*

2020

**Proceedings of the
International Conference on**
**New Music Concepts
Inspired Education and
New Computer Science Generation**

Vol. 7

STUDIO
MUSICA

Accademia Musicale Studio Musica

International Conference on New Music Concepts
Inspired Education and
New Computer Science Generation

Proceeding Book
Vol. 7

Accademia Musicale Studio Musica
Michele Della Ventura
Editor

Preface

This volume of proceedings from the conference provides an opportunity for readers to engage with a selection of refereed papers that were presented during the International Conference on New Music Concepts, Inspired Education and New Computer Science Generation. The reader will sample here reports of research on topics ranging from a diverse set of disciplines, including mathematical models in music, computer science, learning and conceptual change; teaching strategies, e-learning and innovative learning, neuroscience, engineering and machine learning.

This conference intended to provide a platform for those researchers in music, education, computer science and educational technology to share experiences of effectively applying cutting-edge technologies to learning and to further spark brightening prospects. It is hoped that the findings of each work presented at the conference have enlightened relevant researchers or education practitioners to create more effective learning environments.

This year we received 57 papers from 19 countries worldwide. After a rigorous review process, 24 paper were accepted for presentation or poster display at the conference, yelling an acceptance rate of 42%. All the submissions were reviewed on the basis of their significance, novelty, technical quality, and practical impact.

The Conferece featured three keynote speakers: Prof. **Giuditta Alessandrini** (Università degli Studi Roma TRE, Italy), Prof. **Renee Timmers** (The University of Sheffield, UK) and Prof. **Axel Roebel** (IRCAM Paris, France).

I would like to thank the Organizing Committee for their efforts and time spent to ensure the success of the conference. I would also like to express my gratitude to the program Committee members for their timely and helpful reviews. Last but not least, I would like to thank all the authors for they contribution in maintaining a high-quality conference and I hope in your continued support in playing a significant role in the Innovative Technologies and Learning community in the future.

March 2020

Michele Della Ventura

Conference Chair

Michele Della Ventura, Accademia Musicale Studio Musica, Treviso, Italy

Keynote Speakers
Giuditta Alessandrini, Università degli Studi Roma TRE, Italy
Renee Timmers, The University of Sheffield, UK
Axel Roebel, IRCAM Paris, France

International Scientific Committee
Patricia Alessandrini, Goldsmiths, University of London, UK
Jioanne Armitage, University of Leeds, UK
Suzanne Aspden, Faculty of Music, University of Oxford, UK
Jean-Julien Aucouturier, IRCAM, Paris, France
Per Bloland, Miami University, Ohio, USA
Jeffrey Boehm, Bath Spa University, UK
David Carabias Galindo, University of Segovia, Spain
Marko Ciciliani, University for Music and Performing Arts Vienna, Austria
Sally Jo Cunningham,University of Waikato, New Zealand
Ching-Hua Chuan, University of North Florida, U.S.A.
Darryl N. Davis, University of Hull, UK
Marlo De Lara, University of Leeds, UK
Elga Dorner, Central European University, Budapest, Hungary
Simon Emmerson, De Montfort University, Leicester, UK
Travis Garrison, University of Central Missouri, USA
Inés María Monreal Guerrero, University of Valladolid, Spain
Duncan Williams, University of Plymouth, UK
Andrew Hankinson, Bodleian Libraries, University of Oxford, UK
Joseph Hyde, Bath SPA University, UK
Wladyslaw Homenda, Warsaw University of Technology, Poland
Orestis Karamanlis, Bournemouth University, UK
Alexandros Kontogeorgakopoulos, Cardiff Metropolitan University, UK
Steven Jan, University of Huddersfield, UK
Tae Hong Park, New York University Steinhardt, USA
Rudolf Rabenstein, University Erlangen-Nuremberg, Erlangen, Germany
Silvia Rosani, Goldsmiths, University of London, UK
Robert Rowe, New York University, USA
Nikos Stavropoulos, Leeds Beckett University, UK
Jacob David Sudol, Florida International University, U.S.A.
Eva Zangerle, University of Innsbruck, Austria

Contents

Learning Tools, Leraning Technologies, Learning Practices

Culture and Music

Poster presentation

New Music Concepts

Intelligent Automation of Secondary Melody Music Generation

Nermin Naguib J. Siphocly, El-Sayed M. El-Horbaty, Abdel-Badeeh M. Salem

Computer Science Department, Faculty of Computer & Information Sciences Ain Shams University, Cairo, Egypt.
nermine.naguib@cis.asu.edu.eg, shorbaty@cis.asu.edu.eg, absalem@cis.asu.edu.eg

Abstract. Generating a secondary melody for a given main melody is challenging and relies on human musical experience more than theoretical rules. The aim of this paper is to propose an intelligent method for automating secondary melody generation using artificial neural networks. Experimental results are presented and discussed highlighting the best practices for minimizing the network error function. When trained on 3000 sample the network was successfully trained reaching a loss value of 0.05 after 2000 iteration.

1   Introduction

How humans learn music is fascinating; as they reach very high levels of musical creativity and harmony. Musical pieces formed of multiple voices or multiple melody lines (polyphonic), sound rich and filling to human ears. A musical piece's main melody is the set of consecutive single notes (tones) forming its main musical line. Secondary melody can be defined as a melody line that accompanies the main melody and in harmony with it.

Since the early days of the invention of computers, it has been tempting to utilize their computational powers in aiding musicians with their musical pieces production. An interesting finding in [1] shows that the first music produced by computers was heard in the early forties on Alan Turing's computer named Manchester. The talented programmer Christopher Strachey [2] succeeded to develop a program that plays Britain's national anthem "God Save the King" on Turing's computer.

In order to teach computers music, machine learning techniques, especially those related to '*artificial neural networks*'(ANNs), has been adopted [3, 4, 5]. ANNs simulate humans' methods of learning music by example[6]; just like human singers who practice singing secondary melodies by listening to numerous musical pieces of main/secondary melody pairs. This motivated us in this paper to study the application of ANNs in generating secondary melody line for a given main melody. We first give a theoretical

background about ANNs and Adam (Adaptive Moment Estimation) optimization training algorithm. Afterwards, we elaborate on our application of ANNs and Adam algorithm for generating secondary melody lines automatically. Finally, we discuss our experimental results and implementation.

The rest of this paper is organized as follows; Section 2 describes the recent work related to our topic. Section 3 gives a theoretical background engaged with our work. Section 4 illustrates our proposed method for secondary melody generation by ANNs and Adam algorithm. Section 5 describes our experiments and the observed results highlighting the tools used in our implementation. Section 6 discusses the previously mentioned results. Finally, in Section 7 we conclude our research and give insight on the probable future work related to it.

2 Related Work

ANNs have been extensively espoused in computer music generation applications. Examples of recent research implementing ANNs in computer music includes that of Yamada et al. [3] who devised a comparison between Bayesian Networks (BNs) and recurrent neural networks (RNNs) in chorale music generation highlighting the strengths and weaknesses of each.

As for chord generation using ANNs, a recent research by Brunner et al. [4] combines two long short-term memory neural networks (LSTM) to produce polyphonic music. In Brunner's system, the first LSTM network was responsible for predicting chord progression and the second generates polyphonic music based on the predicted chord progression. On the other hand, Nadeem et al. [5] developed a system that concurrently generates both; melodies alongside their accompanying chords. Their system trains two neural networks in parallel; one for learning the notes and the other for learning the chords. The outputs from both networks are combined through a dense layer followed by a final LSTM layer. These applications mainly rely on unsupervised learning for training the network whereas [5] combines both supervised and unsupervised learning. In our work we adopt supervised learning through a multilayer perceptron neural network which is trained by "Adam" algorithm.

3 Theoretical Background

In this section we give a brief theoretical background on ANNs and Adam optimization algorithm.

Artificial Neural Networks Overview

ANNs simulates the biological neural system that the brain controls. A biological neural system is formed of interconnected neuron cells, each of which process information. Just like biological neuron, an artificial neuron has inputs, a node (body), weights (interconnections) and an output. The interconnections between nodes have weights on them that affect the transfer rate of the input signal.

The architecture of a neural network or how neurons are organized in the network is called "*network topology*". In this work we are concerned only with *multilayer feedforward* network architecture. A multilayer perceptron has one or more layers of nodes between the input and the output layer. Each neuron in one layer of a multilayer perceptron is connected to all the neurons in the following layer; i.e. a multilayer perceptron is fully-connected. Feedforward networks are those in which the signal flows from the input towards the output layer.

ANNs are "*trained*" to solve problems, or in other words they *learn*. In our work we are concerned with "*supervised learning*" in which the network has a desired output that is compared with the network output then the network weights are updated accordingly. This process is achieved by the "*backpropagation algorithm*".

The backpropagation algorithm is explained in details in [6]; its main idea is that the input is propagated forward in the network from the input to the output layer, then the error between the network's output and the desired output is propagated backward from the output layer towards the input layer, updating the network weights during the process. Equation 1 is the weight update equation during backpropagation. Such that w_{jk} is the weight between the kth and j^{th} layers of the network. and Δw_{jk} is calculated by Equation 2 where α is called the learning rate; representing the update rate or step size. σ_k is the error propagated from the previous layer k (moving backward). At the output layer, σ_k is in terms of the difference between the output and desired output. However, in the hidden layers, σ_k is in terms of the weighted sum of the errors from the previous layer. Lastly, z_j is the output of the nodes in the jth layer.

$$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk} \tag{1}$$

$$\Delta w_{jk} = \alpha \sigma_k z_j \tag{2}$$

What Equation 1 does is that it calculates the weights gradients in order to perform "gradient decent" [7] which is the most common optimization algorithm adopted in neural networks. It aims to minimize an objective function by updating its parameters in the opposite direction to its gradient.

Adam Optimization algorithm

Adam (Adaptive Moment Estimation) recently proposed by Kingma and Jimmy [8] is a gradient-based algorithm. As stated in [7] "*Adam behaves like a heavy ball with friction, which thus prefers flat minima in the error surface*". In Adam, adaptive learning rates are calculated for each weight. Adapting per-weight learning rates in Adam are in terms of the rate of change of the gradients for the weight through the average of their recent magnitudes. Equations 3 and 4 show that the algorithm calculates an exponential moving average of the gradient (m_t) and the squared gradient (v_t), such that the parameters β_1 and β_2 control the decay rates of these moving averages.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)g_t \tag{3}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)g_t^2 \tag{4}$$

Throughout training with Adam, the network weights are updated in mini-batches rather than being updated after each sample, which is more efficient for large datasets.

4 A Proposed Method for Secondary Melody Generation using Artificial Neural Networks and Adam Algorithm.

Since our problem is based on learning by example (training on ready-made main/secondary melody pairs), we adopted supervised learning for our network. The learning process is mainly to map group of notes (those of the main melody) to another corresponding group of notes (those of the desired secondary melody). This learning process trains the network to derive the relationship between the main and secondary melody without stating musical rules – thus, simulating human "auditory learners". Since our problem is non-linear, we chose the multilayer perceptron. Adam optimization algorithm was the most suitable for our problem due to its efficiency with large training data due to using mini-batches. Moreover, the adaptive learning rate is more suitable for music application of ANNs because this is more abiding to music nature that the distances between notes is not fixed.

5 Experimental Test and Results

We implemented our application using Python 3.7 (Anaconda 2019.10 distribution) under Linux (Xubuntu 18.04 LTS). We used the machine learning python library "*scikit-learn*" 0.22.1 for the ANN implementation – with Adam optimization. We used the standard MIDI format which stands for Musical Instrument Digital Interface. MIDI is a data communications protocol. MIDI data is not an audio signal, but it is rather formed of instructions only transmitted in the form of binary numbers. The creation of the actual audio signal is left to the receiving device.

We conducted our experiments on a dataset of 78 MIDI files from [9] which is a library of four voices carols. Our dataset is formed of files of the single line melodies and their corresponding files of the Tenor or Alto lines, whichever is closer to representing a secondary melody – this decision is done manually by human listeners. We named the files such that each two related melodies differ only in the first letter. Our system scans each file (*f*) in parallel with its corresponding secondary melody file (*fs*) with a window size *ws*. Each *ws*-sized group of notes from *f* and their corresponding group from *fs* are kept in pairs in memory. Each pair represent a sample that will be introduced to the network (the notes read from *fs* represent the desired output of the network). The 78 files of our dataset produced about 3000 sample. We achieved comparable results for window sizes 16 and 32; smaller values were slowly converging, and higher values exceeded the files sizes.

# Hidden Layers	# Epochs	Min Error
16	2000	4.527
200	2000	3.209
1000	2000	0.958
2000	2000	0.242
3500	2000	0.072
3500	1000	0.138
3500	500	0.435
4000	2000	0.050

As follows we list the training results observed by setting different parameters to our network setting the window size to 16. Table 1 shows the different values for the number of neurons in the hidden layer in the first column and the number of epochs in the second column; which is how many times the samples are introduced to the network during training. The last column is the minimum error (loss) achieved by the network at each set-up. Figure 1 portrays six graphs for the error per epochs. The x-axis for all of them is the number of epochs and the y-axis is the error value. The number of hidden layers for graph (a), (b), and (c) is fixed to 3500 neurons. Whereas the number of epochs for graphs (d), (e), and (f) is fixed to 2000. The number of epochs for graphs (a), (b), and (c) is 500, 1000, and 2500 respectively. The number of hidden layers for graphs (d), (e), and (f) are 16, 200, and 3500 neurons respectively.
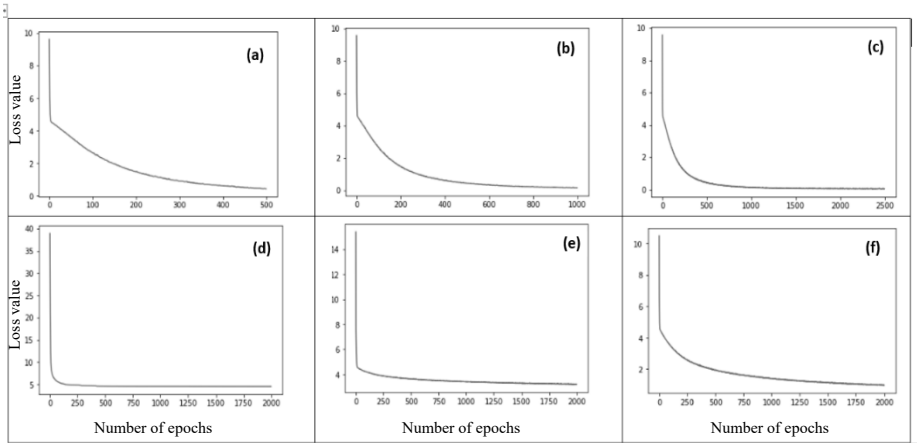

Fig. 1. Training loss graphs relevant to the network parameters changes.

6   Discussion

From Table1 and Figure 1, it is shown that the general behavior of our network is that as the number of hidden layers and epochs increase, the overall performance gets better. However, the performance enhancement becomes minimal after the values 2500 for the epochs and 3500 for the number of hidden layers (compare the third row in Table 1 to the eighth row). Setting the number of hidden layers to be less than or equal the number of inputs - which is equals to the window size – results in the minimum convergence rate, this is shown in Graph (d) where the maximum error rate is 40 at the first epoch and around 4.5 between the epochs 1500 - 2000. Graph (a) is the second worse loss graph because training with 500 epochs was not enough to converge. Generally speaking, the effect of increasing the number of hidden layer neurons on our network's performance is greater than that of increasing the number of epochs – provided that a suitable number of hidden layers is chosen.
We tested our system with some melodies and asked human listeners to judge the quality of the generated secondary melodies and the feedback was promising.

7   Conclusion and Future Work

In this paper we proposed a methodology for automating the generation of secondary melodies intelligently using ANNs. We emulate the way of human auditory music learners who learn music by example and practice. We first highlighted some of the recent research related to our work, we then gave a short relevant theoretical background about ANNs; architecture, learning, and training rule. In addition, we summarized the gradient decent rule and the Adam optimization algorithm. We explained our application for automatically generating secondary melody utilizing ANNs highlighting the design choices and the reasons behind them. We discuss our results that shows the best network set-up and parameters values obtained. Our work can represent a starting point to various works in the future. Future work includes combining other machine learning algorithms with neural networks for achieving results that are even closer to human generated secondary melodies. Our application can be fit as a module in a bigger project for generating a full composition with multiple components and accompaniments.

References

[1]   B. Copeland, J. Long (Eds.), Turing and the History of Computer Music, In: Floyd J., Bokulich A. (eds) Philosophical Explorations of the Legacy of Alan Turing. Boston Studies in the Philosophy and History of Science, vol 324, Springer, Cham, 2017.

[2]   N. Foy, The word games of the night bird (interview with Christopher Strachey), Computing Europe (1974) 10-11.

[3]   T. Yamada, T. Kitahara, H. Arie, T. Ogata, Four-part harmonization: Comparison of a Bayesian network and a recurrent neural network, in: M. Aramaki, M.

E. P. Davies, R. Kronland-Martinet, S. Ystad (Eds.), Music Technology with Swing, Springer International Publishing, Cham, 2018, pp. 213-225.

[4]   G. Brunner, Y.Wang, R.Wattenhofer, J. Wiesendanger, Jambot: Music theory aware chord based generation of polyphonic music with LSTMs, in: 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI), 2017, pp. 519-526.

[5]   M. Nadeem, A. Tagle, S. Sitsabesan, Let's make some music, in: 2019 International Conference on Electronics, Information, and Communication (ICEIC), 2019, pp. 1-4.

[6]   L. Fausett (Ed.), Fundamentals of Neural Networks: Architectures, Algorithms, and Applications, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1994.

[7]   S. Ruder, "An overview of gradient descent optimization algorithms", ArXiv(2016), arXiv:1609.04747.

[8]   D. P. Kingma, and B. L. Jimmy, "Adam: A Method for Stochastic Optimization", he 3rd International Conference for Learning Representations, San Diego, 2015. arXiv:1412.6980.

[9]   Royal Free Music Society Choir. 100 Carols for Choirs: "https://royalfreemusicsociety.org.uk/100-carols-for-choirs". Last accessed on January 12, 2020.

This book presents a collection of selected papers that present the current variety of all aspect of the research at a high level, in the fields of music, education and computer science. The book meets the growing demand of practitioners, researchers, scientists, educators and students for a comprehensive introduction to key topics in these fields. The volume focuses on easy-to-understand examples and a guide to additional literature.

Michele Della Ventura, editor

New Music Concepts, Inspired Education, Computer Science

Revised Selected Papers