Generación automática de notas musicales mediante autocodificadores variacionales condicionales
Automatic generation of music notes through conditional variational autoencoders

Trabajo de Fin de Grado

Curso 2024–2025

Autor

Pablo García López

Directores

Miguel Palomino Tarjuelo

Jaime Sánchez Hernández

Grado en Ingeniería Informática

Facultad de Informática

Universidad Complutense de Madrid

Generación automática de notas musicales mediante autocodificadores variacionales condicionales
Automatic generation of music notes through conditional variational autoencoders

Trabajo de Fin de Grado en Ingeniería Informática

Autor
Pablo García López

Directores
Miguel Palomino Tarjuelo
Jaime Sánchez Hernández

Convocatoria: *Junio 2025*

Grado en Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid

Resumen

Generación automática de notas musicales mediante autocodificadores variacionales condicionales

Un resumen en castellano de media página, incluyendo el título en castellano. A continuación, se escribirá una lista de no más de 10 palabras clave.

Palabras clave

Deep Learning, Conditional Variational Autoencoders

Abstract

Automatic generation of music notes through conditional variational autoencoders

An abstract in English, half a page long, including the title in English. Below, a list with no more than 10 keywords.

Keywords

Deep Learning, Conditional Variational Autoencoders

Contents

List of figures

List of tables

Chapter 1

Introduction

"Predicting the future isn't magic, it's Artificial Intelligence"
— Dave Waters

1.1. Motivation and Objectives

In recent years, deep learning has revolutionized generative tasks in fields like image synthesis, natural language processing, and audio production. Within music, research has generally split into *symbolic* approaches (focusing on note events, pitches, and durations in formats like MIDI) and *non-symbolic* approaches (focusing on raw audio waveforms or spectrograms).

Commercial digital audio workstations (DAWs) and synthesizers already allow users to generate audio with great precision. However, these are often not driven by *deep-learning*–based methods. Moreover, there is a compelling interest in exploring new audio possibilities achieved by *learned* latent representations, e.g., timbres or articulations that might not exist in standard synthesizer libraries.

This Bachelor's Thesis therefore focuses on the implementation of a **Conditional Variational Autoencoder (CVAE)** for directly generating non-symbolic musical notes. Through conditioning, guiding the network with desired musical parameters—such as approximate pitch, instrument type, or intensity—should be feasible. The aim is to combine the *flexibility* and *freshness* of learned audio representations with the *usability* of a straightforward interface that lets users "play" with different configurations to generate sounds. While the results may not surpass the polish or versatility of commercial synthesizers, such a model can reveal new pathways for interactive sound design and serve as a research-driven educational tool.

In any way, we would like this project to serve as an introduction and guide for students or anyone interested in the use of deep learning in music. While we do not assume extensive knowledge from the reader, we also will not go into excessively detailed explanations in order to keep the text accessible.

1.2. Work Plan

This section describes the work plan to follow in order to achieve the objectives outlined in the previous section.

(Pongo aquí esto mejor yo creo) We will need to define a metric for the loss function, in order to quantify how good the sample generation provided by the CVAE is.

Chapter 2

State of the Art

In this chapter, we aim to first provide a brief overview of the evolution of algorithmic composition and, secondly, explore non-symbolic (i.e., low-level) music generation more in depth.

2.1. Brief history of algorithmic composition

Algorithmic composition is the process of using some formal process to make music with minimal human intervention (Alpern, 1995) and can be divided into two main categories: *non-computer-aided* and *computer-aided* methods.

2.1.1. Non-computer-aided methods

Algorithmic composition dates back thousands of years. In Ancient Greece, philosophers such as Pythaforas viewed music as fundamentally linked to mathematics, believing that musical harmony reflected universal order (Simoni, 2003). These ancient Greek "formalisms," however, are rooted mostly in theory, and their strict application to musical performance itself is probably questionable (Grout and Palisca, 1996). Therefore, it can't really be said that Ancient Greek music composition was purely algorithmic in the sense we have defined it, but it undoubtably set the path towards important formal extra-human processes.

In the Renaissance and Baroque, algorithmic methods became more explicit through forms like the canon, where composers, like Johann Sebastian Bach, created strict rules dictating how single melodies are to be imitated by multiple voices at different times.

A famous Classical-era example is Mozart's *Musikalisches Würfelspiel* ("Dice Music") in which musical phrases were randomly assembled by dice rolls, explicitly employing chance-based algorithmic composition (Maurer, 1999).

The 20th century introduced more complex algorithmic techniques through serialism, where composers like Arnold Schoenberg and Alban Berg employed systematic tone-row matrices (see Figure 2.1 to structure their compositions through fixed rules. Composers such as John Cage and Karlheinz Stockhausen later incorporated chance and probabilistic methods, further extending the tradition of algorithmic music before the advent of computers (Simoni, 2003).

	I_0	I_{10}	I_3	I_4	I_2	I_1	I_{11}	I_9	I_8	I_7	I_5	I_6	
P_0	E♭	D♭	G♭	G	F	E	D	C	B	B♭	A♭	A	R_0
P_2	F	E♭	A♭	A	G	G♭	E	D	D♭	C	B♭	B	R_2
P_9	C	B♭	E♭	E	D	D♭	B	A	A♭	G	F	G♭	R_9
P_8	B	A	D	E♭	D♭	C	B♭	A♭	G	G♭	E	F	R_8
P_{10}	D♭	B	E	F	E♭	D	C	B♭	A	A♭	G♭	G	R_{10}
P_{11}	D	C	F	G♭	E	E♭	D♭	B	B♭	A	G	A♭	R_{11}
P_1	E	D	G	A♭	G♭	F	E♭	D♭	C	B	A	B♭	R_1
P_3	G♭	E	A	B♭	A♭	G	F	E♭	D	D♭	B	C	R_3
P_4	G	F	B♭	B	A	A♭	G♭	E	E♭	D	C	D♭	R_4
P_5	A♭	G♭	B	C	B♭	A	G	F	E	E♭	D♭	D	R_5
P_7	B♭	A♭	D♭	D	C	B	A	G	G♭	F	E♭	E	R_7
P_6	A	G	C	D♭	B	B♭	A♭	G♭	F	E	D	E♭	R_6
	RI_0	RI_{10}	RI_3	RI_4	RI_2	RI_1	RI_{11}	RI_9	RI_8	RI_7	RI_5	RI_6	

Figure 2.1: Serialism matrix

2.1.2. Computer-Aided Methods

The advent of computers in the mid-20th century significantly advanced algorithmic composition, introducing computational techniques that expanded creative possibilities. Early pioneers like Lejaren Hiller and Leonard Isaacson composed the *Illiac Suite* (1957), one of the first pieces generated entirely by computer algorithms (Hiller and Isaacson, 1959). They utilized a generator/modifier/selector framework, where musical materials were algorithmically created, modified, and selected based on predefined rules (Maurer, 1999).

Composer Iannis Xenakis introduced *stochastic music*, employing probabilistic methods to generate musical structures. For instance, in his work *Atrées* (1962), Xenakis used probability distributions and random number generators to determine musical elements (Xenakis, 1992).

Computer-aided algorithmic composition can be categorized into three main approaches:

1. Stochastic systems: they incorporate randomness, ranging from simple random note generation to complex applications of chaos theory and nonlinear dynamics (Nierhaus, 2009).

2. Rule-Based systems: these utilize explicitly defined compositional rules or grammars, similar to earlier non-computer methods like the Renaissance canons or serialist compositions we have talked about. Notable examples include William Schottstaedt's automatic species counterpoint program and Kemal

Ebcioglu's CHORAL system, which generate music based on historical compositional rules (Cope, 1991).

3. Artificial Intelligence systems: these systems extend rule-based methods by allowing a computer to develop or evolve compositional rules autonomously. David Cope's Experiments in Musical Intelligence (EMI) exemplifies this approach, analyzing existing compositions to create new music emulating specific composers' styles (Maurer, 1999).

2.2. Non-symbolic music generation

In Section 2.1 we gave an overview of historical algorithmic composition along with its two main branches: non-computer-aided and computer-aided methods, which largely focus on *symbolic* or high-level approaches. In this section, however, we turn our attention to *non-symbolic* music generation, where the emphasis is on generating and shaping audio signals directly.

We begin with an overview of foundational digital synthesis systems, which provided the bedrock for modern audio generation. We then discuss recent AI-based approaches, including various deep-learning architectures capable of producing music at the waveform (or spectrogram) level. Although this thesis aims to ultimately employ a conditional variational autoencoder for generating musical notes, understanding the broader ecosystem of audio-focused methods places our work in context.

2.3. Traditional Digital Synthesis Systems

2.3.1. Additive Synthesis

Additive synthesis constructs timbre by summing multiple sinusoidal partials at different frequencies, amplitudes, and phases. It directly parallels Fourier analysis in decomposing signals into sinusoidal components.

2.3.2. Subtractive Synthesis

Subtractive synthesis starts with a spectrally rich waveform (e.g., sawtooth, square) and removes components via filters (low-pass, high-pass, band-pass).

2.3.3. Frequency Modulation (FM) Synthesis

Popularized in the 1970s by John Chowning, FM synthesis uses one oscillator (the modulator) to vary the frequency of another (the carrier), producing complex sideband spectra.

2.3.4. Other Approaches

Here I want to briefly comment, Granular Synthesis, Physical Modeling, and Spectral Modeling.

2.4. Modern AI-Driven Non-Symbolic Music Generation

Unlike traditional systems based on handcrafted signal-processing algorithms, **deep learning** methods for non-symbolic music generation learn representations directly from data. They can produce raw audio or time-frequency representations (e.g., spectrograms).

2.4.1. Waveform Modeling Approaches

WaveNet, introduced by van den Oord et al. (2016), is a generative model employing dilated convolutions to predict audio samples. Although originally targeted at text-to-speech, it was adapted for music textures and timbral generation.

SampleRNN uses a hierarchy of recurrent neural networks to model raw audio samples at multiple time scales. It can capture both short-term details and longer-term musical structure.

Generative Adversarial Networks (GANs) have also been applied. *GANSynth* from Google Magenta synthesizes audio in the frequency domain and can produce notes with controllable pitch and timbre.

2.4.2. Latent Space Modeling (Autoencoders)

Autoencoders compress and reconstruct data by learning latent representations.

2.4.2.1. Variatonal Autoencoders (VAEs)

Variational Autoencoders add probabilistic encoding, enabling smoother latent spaces that can be sampled or interpolated.

2.4.2.2. Conditional VAEs (CVAEs)

By adding conditional inputs (e.g., pitch labels, instrument type), CVAEs can generate audio with desired attributes.

Chapter 3

Basic musical concepts

[Explain basic musical concepts and put émphasis on those that the dataset uses]

Chapter 4

Introduction to Deep Learning and Conditional Variational Autoencoders

[Here I want to give a basic introduction: what deep learning is, what we understand by learning, training a model, the importance of the dataset (and its partitions), loss functions, etc. I do not intend to go into much detail but rather provide the main intuitions.]

[Then, I want to focus on VAEs, starting by explaining what Autoencoders are, what Variational Autoencoders are, and Conditional Variational Autoencoders (architecture, uses, problems, etc.). I also want to succinctly explain the mathematics behind them.]

Conclusions and Future Work

Conclusions and future lines of work. This chapter contains the translation of Chapter **??**.

Bibliography

Alpern, A. Techniques for algorithmic composition of music. `http://alum.hampshire.edu/~adaF92/algocomp/algocomp95.html`, 1995. Accessed: March 2024.

Cope, D. *Computers and Musical Style*. A-R Editions, 1991.

Grout, D. J. and Palisca, C. V. *A History of Western Music*. W. W. Norton & Company, New York, 5th edn., 1996.

Hiller, L. and Isaacson, L. *Experimental Music: Composition with an Electronic Computer*. McGraw-Hill, 1959.

Maurer, J. A. A brief history of algorithmic composition. 1999. Retrieved from `https://ccrma.stanford.edu/~blackrse/algorithm.html`.

Nierhaus, G. *Algorithmic Composition: Paradigms of Automated Music Generation*. Springer, 2009.

Simoni, M. *Algorithmic Composition: A Gentle Introduction to Music Composition Using Common LISP and Common Music*. Michigan Publishing, University of Michigan Library, Ann Arbor, MI, 2003.

Xenakis, I. *Formalized Music: Thought and Mathematics in Composition*. Pendragon Press, 1992.

Appendix A

Título del Apéndice A

Los apéndices son secciones al final del documento en las que se agrega texto con el objetivo de ampliar los contenidos del documento principal.

Appendix B

Título del Apéndice B

Se pueden añadir los apéndices que se consideren oportunos.