Generación de componentes para la creación y evaluación de poemas musicales: Módulo de componentes para la generación musical

Trabajo de Fin de Grado
Curso 2019–2020

Autor
Guillermo Villeta Torres

Director
Gonzalo Méndez Pozo
Pablo Gervás

Grado en Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid

Generación de componentes para la creación y evaluación de poemas musicales: Módulo de componentes para la generación musical

Trabajo de Fin de Grado en Ingeniería Informática
Departamento de Ingeniería del Software e Inteligencia Artificial

Autor
Guillermo Villeta Torres

Director
Gonzalo Méndez Pozo
Pablo Gervás

Convocatoria: *Febrero2020*
Calificación: *Nota*

Grado en Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid

23 de enero de 2020

Autorización de difusión

El abajo firmante, matriculado en el Grado en Ingeniería en Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Grado: "Generación de componentes para la creación y evaluación de poemas musicales: Módulo de componentes para la generación musical", realizado durante el curso académico 2019 - 2020 bajo la dirección de Gonzalo Mendez Pozo y Pablo Gervás en el Departamento de Departamento de Ingeniería del Software e Inteligencia Artificial, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Guillermo Villeta Torres

23 de enero de 2020

Dedicatoria

A Mor, por estar en las buenas y en las malas.
En Madrid, en Barcelona o en el fin del Mundo.

Agradecimientos

Gracias a mi familia, por el apoyo y la comprensión recibidos durante los últimos años durante mi tiempo en la Universidad.

Gracias a los directores de este proyecto, Gonzalo y Pablo, por su apoyo incondicional y la ayuda prestada, sus consejos y sus guías durante todo este tiempo.

Gracias a Marisa y Carlos, compañeros de aventura desde el primer hasta el último día, sin vosotros no hubiera sido lo mismo. Gracias por hacerme disfrutar del viaje y poder compartirlo con vosotros.

Hace casi cinco años entré en la Facultad de Informática sin saber que en ella descubiría mi pasión, gracias a todas aquellas personas que se han cruzado en mi camino durante todo el tiempo que he tenido la suerte de estar en la Facultad de Informática.

Dont cry because it is over, smile because it happened
Dr Seus

Resumen y Preámbulo

El presente trabajo consta del diseño e implementación de un sistema de generación musical basado en técnicas utilizadas en el campo de la Inteligencia Artificial. Este TFG se engloba dentro de un proyecto mayor en el que se generan música para poemas también autogenerados y se lleva a cabo un estudio posterior de su empaste. Los módulos pertenecientes a las secciones de generación de poemas y evaluación del empaste, fueron presentados por Mª Luisa Quiroga y Carlos Martín Testillano en un TFG anterior el año 2018-2019 bajo el título "Generación de componentes para la creación y evaluación de poemas musicales".

La composición musical ha sido un campo de investigación que se ha enfocado desde diferentes ángulos. Desde la propia teoría musical, hasta la psicología, han estudiado la evolución y el efecto de la música y sus patrones compositivos.

En este trabajo, se quiere dar un enfoque diferente a la composición musical y, ya que ésta ha estado siempre vinculada a un factor humano, durante el transcurso del documento hablaremos de "generación musicalz no de çomposición musical". Esta razón se debe a que al no haber una persona que sea directamente partícipe en el proceso compositivo, ni que tome decisiones activamente, se pasa a considerar que las canciones surgidas son generadas y no compuestas.

El sistema, por tanto, es un sistema capaz de estudiar los patrones compositivos de las melodías que le son facilitadas. Dicho estudio permite la generación posterior de una melodía completamente nueva e independiente de las canciones facilitadas pero manteniendo algunas de sus características a partir de su aprendizaje.

Durante el desarrollo del proyecto se han estudiado y probado diferentes técnicas y métodos de Inteligencia Artificial relacionados con la generación musical y se han evaluado los resultados de las técnicas utilizadas atendiendo a diferentes criterios. Adicionalmente, se ha estudiado la posibilidad de recrear técnicas compositivas clásicas.

Palabras clave

Música, Generación, Inteligencia Artificial, Modelos de Markov, Redes Neuronales, Composición

Abstract and Preamble

The present work it is focused on the design and implementation of a music generation system based on techniques used in the Artificial Intelligence field. This TFG is embedded on a larger project designed to generate music to accompany, also generated, poems. A study of their matching is also performed. The modules in charge of generating the poems and studing their matching with the generated music, are covered in another TFG developed by Mª Luisa Quiroga and Carlos Martín Testillano on the preceding year. This work was titled «Generación de componentes para la creación y evaluación de poemas musicales».

Music composition has been a field of research that has been approached from very different prospectives. From music theory, upto psicology have studied the evolution and effect of music and its composition patterns.

In this project, a different approach has been taken and, due to the fact that music composition has always been formed by a human component, during the development of the project we will refer to "music generation" rather than "music composition". The reason for it is because, as no person is involved during the generation process, we will consider the resulting pieces generated and not composed.

The resulting system is, therefore, a system capable of studying the composition patterns followed by given melodies. This study allows the later generation to come up with a brand new piece, but maintaining some of the characteristics found on the input song throughout its learning.

During the development of the project, two main techniques have been studied for music generation: Markov Chains and Neural Networks. A posterior evaluation of the resulting pieces generated with both techniques has

been performed. Furthermore, along with the generation of the music, some techniques found in music composition have been recreated using technology to give pieces a higher degree of musicality.

Keywords

Music, Generation, Artificial Intelligence, Markov Models, Neural Networks, Composition

Contents

List of figures

List of tables

Chapter 1

Introduction

"Music begins where the possibilities of language end"
— Sibelius

1.1. Background

This TFG is embedded on a larger project designed to generate music to accompany, also generated, poems. A study of their matching is also performed. The TFG that cover the modules related to Lyrics generation and music-lyrics matching were presented last September by Maria Luisa Quiroga and Carlos Martín Testillano in a TFG titled «Components generation for the creation and evaluation of musical poems»

That TFG had as aim to compose and evaluate the matching capabilities of both music and lyrics. That TFG took, as part of the input to compose those models, the output of the present TFG, that is the composed music.

1.2. Objective

My aim in this project is to study the capacity of computers to compose music using different techniques belonging to the AI field. These techniques were chosen due to their versatility, the ductility of the models and the tools. Moreover, I consider AI to be an interesting research field due to the exponential growth and the trending emerging technologies.

The scope of the project, therefore, is to design and implement different models, using different techniques, able to generate music and the later

evaluation of the result of each one of the models.

Thus, the project will focus on the generation and evaluation of music and the comparative study of the possibilities offered by the field of music generation through Artificial Intelligence methods.

In the same way, the development of this project will allow me to deepen the knowledge of the structures and models used in AI and at the same time, to combine it with music.

Furthermore, it is intended that this project can be extended and reused by possible next TFGs or serve as an inspiration to continue finding utility to the generative models beyond the mere computational potential of these models.

1.2.1. Motivation

Music is one of the unique ways we humans communicate and transmit ideas and feelings. On the other hand, the computational capacity of today's computers and Artificial Intelligence models allow us to create systems capable of imitating human skills, both in terms of the results obtained and the learning process. At the same time, studying the internal processes of these systems and the variants that lead these systems to generate better results.

In order to consider what is good or bad music, there are varied and diverse criteria within the music theory researchers themselves. That is why during the development of this project, when evaluating the pieces resulting from the composition, we will not base ourselves on objective, parameterized studies and characteristics that try to approach such comparative studies.

Additionally, and through the empirical method of generating music with different techniques, we will try to parameterize certain aspects involved in the generation of music in order to compose better pieces as well as to improve the learning capacity of the models used.

Since music serves, among other things, to transmit feelings and each piece of music carries with it a sentimental degree, a study will be carried out which aims to analyse the association and relationship between feelings and the pieces of music. For this study, the resources used by the musical composers to transmit feelings (fugues, mishaps, dialogues,...) will be taken into account and the factors intervening in this aspect will be developed. In order to recreate these compositional techniques, the differential factors

of each of the techniques and when they are present in the musical pieces will be studied. The aim of this study is to see in which situations these techniques are useful when expressing feelings.

Through this study, the aim is to carry out the parameterization of the techniques in order to recreate them later. To do so, the variables that make possible the musical composition associated with a certain feeling will be parameterized, making possible the enrichment of the musical pieces generated.

The final objective is to try to get the Artificial Intelligence systems used to compose music, not to approach human compositions, but to be able to provide the composed pieces with feeling and meaning beyond the correct succession of notes and silences. This factor, together with the possibilities offered by the models used, will be the differential factor in making this project an approach to music generation through the unification and use of different compositional techniques belonging to the field of music theory and Artificial Intelligence.

Chapter 1

Introducción

"La música comienza donde termina el lenguaje"
— Sibelius

1.1. Antecedentes

Este proyecto se engloba dentro del proyecto «Generación de componentes para la creación y evaluación de poemas musicales», Trabajo de Fin de Grado de María Luisa Quiroga y Carlos Martín Testillano, presentado el pasado mes de Septiembre.

Dicho proyecto tenía por objetivo la composición y evaluación de poemas musicales y se centraba en generar poemas y evaluar el nivel de empaste con la música generada. En el presente TFG se centra la generación de música que servirá de empaste a los poemas. El resto de módulos se han presentado en el TFG nombrado anteriormente.

1.2. Objetivo

Mi objetivo en este proyecto es el estudio de la capacidad de los ordenadores de componer música a través del uso de diferentes técnicas y métodos. Dichos métodos pertenecerán al campo de la Inteligencia Artificial. La elección de dichas técnicas viene dado por la versatilidad de las herramientas y los modelos que dichas técnicas ofrecen. De igual manera, considero interesante el estudio de la IA dado el auge y las posibilidades que ofrece este campo para el estudio de la generación musical.

El alcance del proyecto por tanto, se basará en la creación de diferentes modelos capaces de generar música y la evaluación de resultados de cada uno de los modelos utilizados acorde a una serie de parámetros establecidos de cara a medir el comportamiento y desempeño de cada uno de los modelos.

De tal manera, el proyecto estará centrado en la generación y evaluación de música y el estudio comparativo de las posibilidades que ofrece el campo de la generación de música mediante métodos de Inteligencia Artificial.

De igual manera, el desarrollo de este proyecto me permitirá profundizar en el conocimiento de las estrucutras y modelos utilizados en IA y al mismo tiempo, aunarlo con la música.

Más aun, se pretende que este proyecto pueda ser ampliado y reutilizado por posibles siguientes TFGs o sirva como inspiración para seguir encontrando utilidad a los modelos generativos más allá del mero potencial computacional de estos modelos.

1.2.1. Motivación

La música es una de las maneras únicas que tenemos los humanos de comunicarnos y transmitir ideas y sentimientos, por otro lado, la capacidad computacional de los ordenadores y modelos de Inteligencia Artificial actuales nos permiten la creación de sistemas capaces de imitiar las habilidades humanas, tano atendiendo a los resultados obtenidos como al proceso de aprendizaje. Al mismo tiempo, estudiar los procesos internos de dichos sistemas y las variantes que llevan a estos sistemas a generar mejores resultados.

De cara a considerar qué es una música buena o mala, existen criterios variados y diversos dentro de los propios investigadores de teoría musical. Es por eso por lo que durante el desarrollo de este proyecto, a la hora de evaluar las piezas resultantes de la composición, no basaremos en estudios y características objetivas, parametrizadas, y que intentan acercarse a dichos estudios comparativos.

De manera adicional, y a través del método empírico de generación de música con diferentes técnicas, se tratará de parametrizar ciertos aspectos intervinientes en la generación de música con el fin de poder componer mejores piezas así como mejorar la capacidad de aprendizaje de los modelos utilizados.

Debido a que la música sirve, entre otras cosas, para transmitir sentimientos y cada pieza musical lleva consigo un grado sentimental, se llevará a cabo un estudio el cual tiene por objetivo analizar la asociación y relación entre sentimientos y las piezas musicales. Para dicho estudio se tendrán en consideración los recursos utilizados por los compositores musicales para la transmisión de sentimientos (fugas, contratiempos, diálogos,...) y se desarrollarán los factores intervinientes en este aspecto. Con el fin de recrear dichas técnicas compositivas se estudiarán los factores diferenciales de cada una de las técnicas y cuando están presentes en las piezas musicales. Con dicho estudio se pretende ver en que situaciones dichas técnicas resultan útiles a la hora de expresar según qué sentimientos.

A través de dicho estudio se pretende llevar a cabo la parametrización de las técnicas con el fin de recrearlas posteriormente. Para ello, se parametrizarán las variables que hacen posible la composición musical que se asocie con un determinado sentimiento haciendo posible el enriquecimiento de las piezas musicales generadas.

El objetivo final es intentar conseguir que los sistemas de Inteligencia Artificial utilizados compongan música, no que se acerque a las composiciones humanas, si no que sea capaz de dotar a las piezas compuestas, de sentimiento y significado más allá de la correcta sucesión de notas y silencios. Este factor, junto con las posibilidades ofrecidas por los modelos utilizados, será el factor diferencial para hacer de este proyecto un acercamiento a la generación musical a través de la unificación y uso de diferentes técnicas compositivas pertenecientes al campo de la teoría musical y a la Inteligencia Artificial.

Chapter 2

State of the Art

2.1. Preface and Background

In order to compose a full musical piece formed by music and lyrics it is good to start by dividing the task into those same pieces of the puzzle: lyrics and music. Furthermore ir order to use that music to compose a musical poem, it must be taken into account the facts and variables that makes a piece of music and some lyrics match together in order to become a proper musical composition.

The question that I try to resolve is a question that has been asked several times and have been revisited several times in the generation and AI field: Can a computer create music?

It was the British composer David Bowie with the system called Verbasizer (Braga, 2016) one of the first modern composers to feed his repertoire with computer generated lyrics. Hans Zimmer is another recognized name in the music field that was interested and saw the potential capabilities computers could offer him in the music composition process, and he contributed on the creation of VJAM[1], a mobile application that helped musicians in the process of music composition.

If we want to find the first relation between computers and music composition, we have to go back to 1961, when the IBM computer IBM 7094 the first computer to sing a song. The song was called «Daisy Bell» and it was composed by Harry Dacre and the programmers that made possible for

[1] http://www.vjamapp.com/

the 7094 to sing were John Kelly and Carrol Lock, but the song was already composed, the computer only was able to play it (Ellis, 2006). Therefore was not a proper composition, even though results were impressing and planted a seed among computer researchers. The task was repeated later, when computer HAL 9000 who also sang this same song. This event was recreated later in 2001 in the film «A Space Odyssey»(Boylan, 1985).

If we dig into the history of proper computer-based music composition, we have to go back to 1951 when the computer scientist Alan Turing, known for decrypting the Nazi code during WWII, created a computer capable of composing little musical pieces. Unfortunately, the computer was destroyed but Jack Copeland and Jason Long (Copeland and Long, 2017) , two researchers from the University of Canterbury in New Zealand back in 2016 were capable of reproducing the computer and some of the pieces Alan Turing composed with that computer.

Computer-based music composition is a wide and studied field, and we can distinguish among the researchers and currents according to the method used and the approach taken in order to create a computer-generated musical piece. In the forthcoming sections I will expose the main currents and techniques used by researchers who dig into this field.

2.2. Composition from previous learning

Most researchers start by studying the way humans compose music and they try to figure out what can we learn from those techniques and processes and how to transmit that knowledge to a computer, as well as factors that can be put into consideration when trying to make a computer come up with musical pieces.

Composing music is nothing else than to arrange a number of limited notes (tones, semitones and silences) one after another. A note is formed by a pitch (tune) and a duration. But there are several factors that arise when deciding what note to put next.

Humans compose music, influenced by their previously learned or heard music. Therefore, the first thought would be to feed the computer with music pieces and find a way to analyse them in order to find common patterns or structure in order to learn from those factors, and then, try to add them into the computer composed pieces. This approach is taken by Carr and

Zukowski (2018), in whose research papers come up with the explanation of the bits and pieces that are taken into consideration in order to build a model that works based on this methodology.

Another important question that can be found in many researchers is the one related to taking into consideration the control over the process when generating music. Controlling the output involves that the user is able to change certain aspects of the configuration of the model itself, or certain aspects directly related to the generation process. If the output can be controlled or driven while music is composed (generation time), the process is called «conditional» (Carr and Zukowski, 2018), otherwise is called «unconditional». Therefore the first type of composition will allow the user to decide or reconfigure certain aspects so that the output adapts to those changes in order for the it to be more pleasant or fit the established requirements.

2.3. Most used techniques and methods

Simplifying music composition, we can define it as the process through which notes (formed by pitch and duration) are put one after the other to form a full musical composition. Of course, there are different variations of the same group of notes, but the way music notes are arranged has a probabilistic variable attached. Of course, there are several techniques being used and points of view from they address this challenge, but they all share this important fact we now want to recall.

A project that shows this fact very well is the one created in PyPoser (William (2018)), which is a project where music is composed merely based on previously set probabilities. Although Oliveira's work was mainly focussed on Poetry Generation with PoeTryMe, or Tra-la-lyrics, both related to computer generation of lyrics, it serves to show there are many possibilities and application of this technique.

The Poetry Generation methods used by Oliveira can illustrate very well the challenges faced by Hugo and the kind of problems I could be facing during the development of my project as well as the possible ways to face and resolve those problems and challenges.

Due to the fact that the field that covers music generation is widely open and diverse we will structure the relevant references to work developed by other investigators in a way that served as inspiration to our work.

Music Composition sets a very big challenge for investigators who wants to develop their abilities as computer and Artificial Intelligence researchers as in order to generate music it is necessary to have a previous knowledge of the field.

When digging into other investigators work, I selected those who were mainly focused on music generation such as Fraçois Pachett (Aucouturier and Pachet, 2003). His work on music generation and research related to music composition and genre classification of music drove us to conclude that one possibility was to use Artificial Intelligence to develop our work. We started studding several techniques applied to music generation such as Markov Chains and Long-Short Term Memory Network (from now on LSTM), that we later complemented with sentiment analysis research. Pachet is has developed several components for classification and composition of music in different genres and musical currents. On his work called "Taxonomy of musical Genres" he investigates about the different facts that makes a piece of music belong to a certain Music Genre.

To classify music pieces and fit them into a determined genre he takes into account some facts as Geographical inclusion, Agregation, Repetition, historical period, etc. We focused on extracting from his work the musical aspect on music generation that made him generate music from an specific genre or classify music into an specific genre using the TFG developed by Caparrini López, Antonio and Perez Molina, Laura that investigated "Genre classification of electronic music"(Antonio Caparrini López, 2017).

The work developed measures the accuracy of music genre classification techniques. This work served us to see the important facts concerning genre classification and therefore those ones to take into account in order to classify music pieces and follow a given pattern. This work served to connect with the sentiment analysis area of our work.

Not necessarily focussed on music composition but in creation of RNN models, it must be recalled the work of Andej Karpathy Karpathy (2015), whose explanation of the internal functioning of Neural Networks was found to be very useful to understand in depth the structure of a Neural Net.

During all the research phase some important aspects to take into account on music generation could be brought up:

2.3.1. Frequently Used Techniques

Hereafter we will expose the main techniques used to create computer models capable of generating music compositions. It is worth recalling that most of these models were born in the mathematical field, and they turned out to be useful in order to create computer-based generated music.

2.3.1.1. Markov Models

This technique is used as the starting point by most researchers that take into account the statistical fact of music composition.

Markov Models are models first introduced by Andrey Andreyevich Markov in 1906 (Basharin et al., 2004). Andreyevich was a Russian mathematician that conceived for the first time this kind of models that, later would give birth to the nowadays known as Markov Chains.

A Markov model is a an automate system arranged in form of state chains. The change among states is only determined by the current state, not taking into account previous states (Kouemou and Dymarski, 2011).

This states are autonomous from each other and the relations among them is stochastic, so the current state is the only factor that is taken into account to find out what the next factor would be.

Putting it into music terms, the probability model studies the chance of a note to be followed by any other given note. This way wee create a branch map where we set correlations among the different notes conceived in a composition.

This way we can start creating a brand new composition starting with the probabilities for a note to be followed by any other given note.

The most common way to put this relation down to paper is by creating a relation table where probabilities among the notes are pictured. The model, starting from the knowledge adquired from the training song, will create a brand new song taking into account the probability table created by the song.

The main advantage of this model is that, musicality factors are nearly always ensured, as long as the piece used for training the model has in it these particular characteristics.

One example of a musician that used Markov Models is Ianis Xenakis, a musician that in 1958 used a Markov Model to create a piece of music called

Analogue (Xenakis, 1958).

2.3.1.2. Neural Networks

A Neural Network is a structure formed by neurons. A neuron is a independent structure that performs an operation on a given input to deliver an output. The Neural Network is formed by one or more neurons arranged in layers. As with Neurons itself, Neural Networks can have one or more layers.

Depending on the way layers and Neurons are arranged, there are several types of neural networks. The Network can receive an input formed by one or many arrays as an input and can deliver one or many arrays as result. As shown in the next figure, extracted from (Karpathy, 2015), depending on the relation among the number of outputs and inputs, we can distinguish among one-to-one, one-to-many, many-to-one or many to many where the first argument indicates the number of inputs and the second argument, the number of outputs.

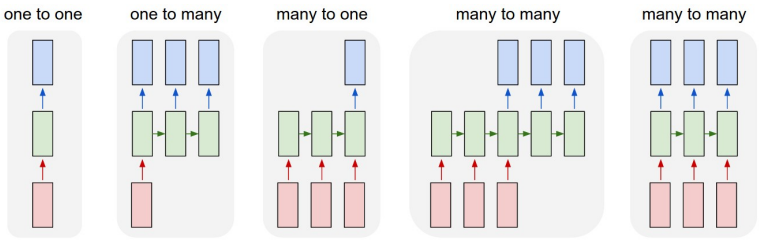


Figure 2.1: Neural Nework types depending on the structure

Among the different layers, the output is forwarded, therefore the output of the previous layer serves as input of the coming layer. At the same time, the output of the later layers can also become part of the input of the previous layers. This property is called «backpropagation». Which turns the Neural Network into a Recurrent Neural Network.

As well as the output of the later layers, the final output of the network can also serve as the input of some of the previous layers or it can be incorporated into the dataset, turning it into the input of the main Neural Network.

This backpropagation capability of the Neural Network, makes a huge impact on the output of the Neural Network, and it is the main difference

among Network and the previously discussed models. This backpropagation property, makes it possible for the Neural Network to learn from previous compossed music and it enriches the output making it more human alike. This capacity to remember previous factors, can allow the Network to compose music with a given structure (which other models did not allow) and it allows for the resulting composition to be more human-alike.

One of the main advantages of this model, as well as the continuous learning factor, it the fact that it also takes into account the evolution of the pieces, therefore, if well configured and properly fed, the Neural Network will allow us to come up with a composition that preserves the structure of a usual song (intro, verse, chorus, verse, chorus, and so on).

The main advantage this kind of models create is the powerful potential of this kind of models when it comes to analyzing music compositions and other kind of structures.

Neural Network models present a big advantage when generating songs are the starting model for music composition, come to embrace and resolve a main issue that appears when using other kind of techniques: the memory factor.

To understand the memory factor we can first start by recalling the structure and coherence along the play of any song. This makes the song to be rithm consistent, but feeling variable. To make things simple, we can understand this as follows: "the songs sounds good, but it sound the same all along".

The way Neural networks tries to resolve this problem is by taking into account, not only the relation among the different notes, or the probability of a note to be followed by any other, but also the main structure of the song and the different probabilities depending on the part of the song we are creating and learning from.

The result of this technique, and what we find to be the biggest advantage, is to create richest musical pieces and more human-alike compositions.

Karpathy wrote a paper back in 2015 titled "The Unreasonable effect of Recurrent Neural Networks" (Karpathy, 2015) on which he relates the functional model of a Neural Network and their possibilities in computational creativity and data handling and generation. He generates Shakespeare text and monologues from training a Recurrent Neural Network.

The way a Recurrent Neural Network works is similar to the Neural Networks found on human brains: the network must receive a set of input data that allows the model to «learn» from it. Iterating over the dataset (a number of specified epochs or iterations), the network is able to determine the weights. The weight is related to the memory of the neural network, as it determines the quantity and/or the quality of the connections among the cells.

The Propagation function is the resulting potential gotten from the interaction of the current neuron with their neighbours.

And finally, the activation functions are the ones that determine the state of liveliness of the neuron. The liveliness of the neuron is the crucial factor that determines if it takes part on the functioning on the network.

The activation function is based on the propagation function and the previous to the interaction.

In the following figure[2] we can see the most common types of activation functions:

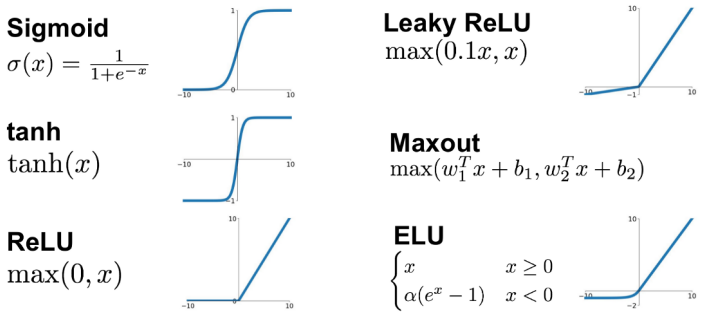


Figure 2.2: Activation function types

Finally, when all previous calculations have been performed, the exit function delivers the result to the neighbouring neurons. If the neuron is inactive, which means that the activation function has not been triggered, the exit function will not deliver any results, turning the neuron into an isolated cell.

[2] https://towardsdatascience.com/complete-guide-of-activation-functions-34076e95d044

2.3.1.3. Main Neural Network Types

As said before in the theoretical part, a Neural Network is formed by neurons arranged in layers. The number of neurons and their attributes can vary depending on the layer and their role in the model. Keras allows us to create a model, stating as parameter of the layers their attributes and the type of NN. The attributes we can specify can be: dropout layers, number of neurons, type of NN, activation function,...

Several types of neural networks can be defined in order to create our model. It is worth recalling that there are nearly infinite types of Neural Networks. This is due to the diversity and variety of possibilities on the design of the network and their behaviour.

Hereafter we specify some of the models that we took took into consideration for the creation of our network and the reasons why we decided to use or discard them in each case:

Feedforward

This is the simplest model of the neural network possibilities. Data travelling through the neural network, only goes in one direction onwards to the output nodes from the input nodes. Its simplicity relies on the position of the neurons and design of the network. It is a first approach to NN

The following figure extracted from (O'Shea and Nash, 2015) shows the typical structure of a Feed Forward Neural Net

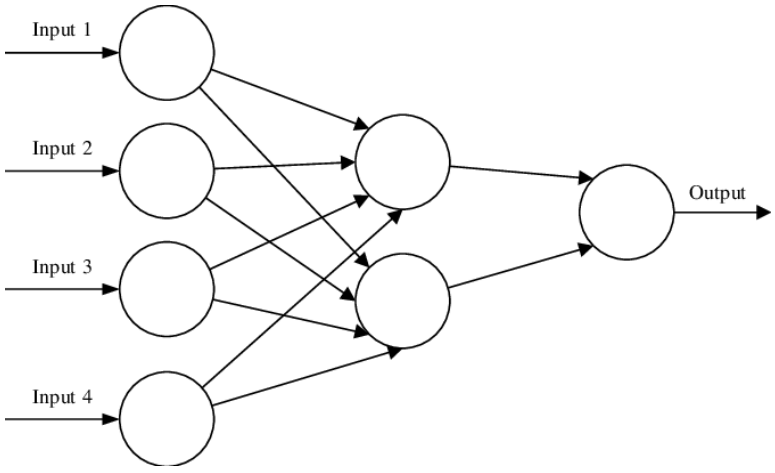


Figure 2.3: Typical structure of a Feed Forward Neural Network

Multilayer Perceptron

The following figure[3] shows the structure of a Multilayer Perceptron Neural Network:

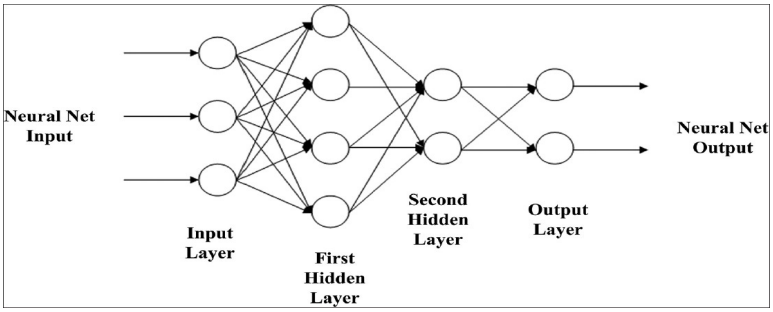


Figure 2.4: Typical structure of a Multilayer Perceptron

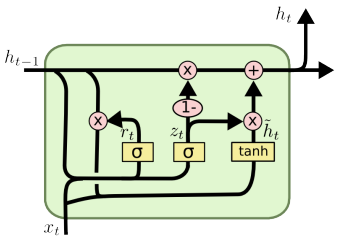
The Multilayer Perceptron (Rosenblatt, 1961) is a neural network that, given a number of inputs, they deliver a series of discrete outputs. This network is normally used, and gives a good result, when the aim is to classify a series of inputs.

The main characteristic of this network is that the number of outputs, as the classes to classify, can not be the same as the number of inputs.

Recurrent Neural Networks - Long-Short Term Memory Networks

The main characteristic of this network is that it exploits the fact of memory-lasting training that the neural networks are expected to deliver.

The following figure[4] shows the usual structure of a LSTM Network:



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Figure 2.5: LSTM typical structure

This is done by registering, and re-feeding the input of the network and its neurons with the results of previous trainings. This model also offers

[3] http://www.jpathinformatics.org/
[4] https://colah.github.io/posts/2015-08-Understanding-LSTMs/

the posibility of back-propagation. Backpropagation is a characteristic that relies on the fact that the output of some of the layers of the network, can be turned into the input of some of the other layers. This is a common feature in Neuran network theory.

This technique gives as a result a better understanding of the structure of the song used for training the network. The result is similar to the one we would get if we chop the song into its parts, and use those several parts as the input of the network. The result of doing this enriches the music with taking into consideration the parts a music composition is divided on, and respects the changes in the feeling and sensations transmitted by music when it is played. A full theory why this types of networks are a good options when it comes to compose music can be found in the article by François Pachet titled "Sampling variation of sequences for structured music generation" (Pachet et al., 2017). A more general approach to this fact can be found in (Karpathy, 2015)

This network, due to the capabilities that it offers, and the nature of its structure, is considered the best possible solution for this project's aim and will become one of the main axes and chosen techniques of the development of the music generation module.

2.3.1.4. Genetic Algorithms

A genetic algorithm, is a computer algorithm that deploys the factor of DNA mixture among two parents. On the DNA structure of a child, we can find common structures found on the DNAs coming from the parents that generated the child.

Brought to music composition, this techniques mixes factors studied in two musical pieces and performs changes into those factors and the combine them to create a brand new musical piece (Horner and Goldberg, 1991).

The most common variations of this factors could be:

- Permutation

  Swapping position and mixing values among the factors to create new relations

- Crossovers

Swapping factors between the parents and keeping others to create a new piece

- Stochastic changes

  By using stochastic pseudo-random changes on the parents to create a new song

Most of the times, this techniques can be combined to deliver a richer musical composition that distanciates and differenciates itself of the original music compositions taken as starting points.

2.3.2. Sentiment Analysis

The last researcher work we would like to recall, and it is probably one of the most important contributors to our work is Gehard Wildmer.

Wildmer, in his work, studies the process of music interpretation and sentiment analysis linked to the physical consequences of music interpretation such as tension, effort, measure, delay, diagnotons, and other aspects (Knees et al., 2007).

Wildmer's work also focuses on expressiveness and sentiment communication over music performance and also relates its work to music genre classification. Due to his deep knowledge over many music facts and technicalities, he is able to relate music composition, creation, interpretation, rhythm patterns and performances with other aspects such as, what he calls "surprising musical discoveries" analysing the effect of music "miss-composition" which is the fact of composing or generating music in a way that is unpredictable to mere hearing sense or it is not composed by following what is considered to be the standard composition flow. (Gouyon et al., 2004)

Lastly, we would like to point out the work developed by Jaime Altozano. He is a young music expert who maintains a YouTube channel dedicated fully to music knowledge spreading which, though the development of this project helped us to see different aspects related to music composition and music theory applied to many well-known music pieces and to take into account theoretical music aspects in order to generate better quality music by our components.

Chapter 3

Tools

In this chapter the different tools and utilities that have been used for the development of the solution will be enumerated and explained, as well as the reasons why these were chosen before others.

3.1. Programming language and libraries

The project has been developed entirely using the Python[1] programming language, mainly due to the wide variety of open source libraries available, as well as the community that it has behind and the fact that AI focus mainly on Python due to its mathematical fundamentals. It is also a language commonly used when working with large amounts of data, so it was determined that it would be positive for both music and lyrics composers.

The environment used was Anaconda[2] because it comprises several utilities that were valuable for the purpose of the development procedure. For designing and debugging functions independently, the *Jupyter Notebook* and *Spider* tools were used because of their simplicity and the convenience for evaluating results in its interface.

The main libraries and platforms used for developing the solution are:

[1] `https://www.python.org/`
[2] `https://www.anaconda.com/`

3.1.1. Music21

Music21[3] is a library developed by a research group in the MIT (Massachusetts Institute of Technology) to provide users, mainly related to the investigation field, an easy way to study and work with music. It was created mainly to work with classical music pieces, but it has a large repertoire of functions and utilities that make it possible to work with any kind of music as long as the input format is supported.

For the evaluation process this was the main utility used because of its simplicity and intuitiveness to study the already composed music. It has also been chosen due to the ability it has to assign lyrics to a given piece, as well as the wide variety of input and output formats it accepts.

3.1.2. Numpy

Numpy[4] is a package for scientific computing with Python. Among other functionalities, it contains advanced computing capabilities, tools for integrating code in other languages, sophisticated functions and the possibility to use it as a container of large amounts of generic data.

3.1.3. Neural Networks (NN)

Several options are available to create a model based on NN. Some of them focus on the statistical and mathematical approach and nature of the solution while other focus on their performing and usefulness capabilities.

3.1.3.1. Theano

Theano[5] is probably the most known and recognizable options of all for NN handling. This is a library that was created to define, optimize, and evaluate mathematical expressions, involving multidimensional arrays.

While it is the most veteran of the options, it is also a very low level implementation, and even though there has been some efforts to turn it into a better and more usable option for beginners, it is still focused on the low level infrastructure.

[3] `https://web.mit.edu/music21/`
[4] `https://numpy.org/`
[5] `https://github.com/Theano/Theano`

3.1.3.2. Tensorflow

Tensorflow[6] is a Python library good for numerical computation, it is supported by a strong community behind it and is the most preferred option among developers. The fact of a strong community developing the library plus the fact of a low level set of operations, it makes tensorflow a good option for own generations of behavioural models as NN.

The main drawback is that, given the nature of the library, it is focused in a variety of low level data manipulation and mathematical transformations.

These two aspects, make Tensorflow a good option for data manipulation, and even though we have performed several attempts and even developed a few models on tensorflow to handle Music, we finally through away the idea due to the amount of code to be developed to get results and the level of detail necessary to configure the library and create the model.

3.1.3.3. Keras

The nature of Keras[7] relies on the high-level approach it takes to the NN definition of the model, as well as the definition of the parts of the NN (layers and neurons). Allowing a high-level approach, let us configure the parameters of the network making it very reliable and reusable, easing the use of the model. Also the available documentation about the Keras API is accessible and well supported making it clear and simple to use.

It runs over Tensorflow and Theano, allowing, if necessary, to go deeper into a low level modification of the structure and the connections of the neurons.

It has a simple and clear documentation and makes the code clear, simple and let us focus on the design of the network careless of the underlying infrastructure.

All the above mentioned facts, makes Keras the best option for NN handling and programming NN structures.

[6] `https://www.tensorflow.org/`
[7] `https://keras.io/`

3.1.4. MIDI

The MIDI format (Musical Instrument Digital Interface) is a simple protocol to communicate synthesizers and other electronic music equipment and transmit music events.

It was developed in 1981 by Dave Smith and Chet Wood of Sequential Systems(Igoudin, 1997). MIDI was quickly embraced by all the major synth manufacturers and led to developments such as microcomputer sequencers, and with them the electronic home studio. Although many attempts have been made to replace it, it is still the industry standard.

Midi files are formed by events, those events can represent music elements. Midi is based on the correlation among the different notes in musical notation and their corresponding frequencies and duration and a number. The following table shows the correlation among music elements and MIDI format file.

Table 3.1: Midi correlation

Clef	Note	MIDI Number	Frecuency
Bass	C3	48	131 Hz
	C♯ 3/D♭3	49	139 Hz
	D3	50	147 Hz
	D♯3/E♭3	51	156 Hz
	E3	52	165 Hz
	F3	53	175 Hz
	F♯3/G♭3	54	185 Hz
Bass and Treble	**C4** (middle C)	**60**	262 Hz
Treble	C♯4/D♭4	61	277 Hz
	D4	62	294 Hz
	D♯4/E♭4	63	311 Hz
	E4	64	330 Hz
	F4	65	349 Hz

3.1.5. Piano Roll

Piano roll is a music storage medium that defines a channel for each of the keys of the keyboard, and, starting with the activation of those channels throughout time, it stores musical pieces. It is based on the music rolls that operated mechanical instruments (Bryner, 2002)

3.1.6. MIDO

Mido[8] is a library used to work and manipulating MIDI events. It is a simple an straightforward method to handle and manipulate MIDI files and events.

This library will be used in the Markov module of the project.

3.1.7. LilyPond

Lilypond[9] is a music engraving program, designed to produce sheet musics from MIDI files. This library will be used to convert the output of the model into a well-formed music sheet.

3.1.8. Musescore

The output of the solution can be presented in multiple formats, one of the formats I want to make available the pieces composed by the system is the classical music Sheet. The principal aim of this fact is to make available and reproducible the outputs of the systems.

For that purpose, musescore allows to change, visualize, reproduce and modify the music created by the components. This will allow me to see and analyse in more depth the variations in music explained in the previous chapters created by the models.

3.2. Hierarchical Data Format: HDF5

HDF5 is an open source technology that allows perform handling operations on large data collections. In this project, hdf5 files will be used to load the weights used in the neural network to generate music.

[8]https://github.com/mido/mido
[9]http://lilypond.org/

Chapter 4

Development approach towards music generation

The following chapter will put down in paper the development of the created models for music generation. Therefore, it contains the techniques used, the final model created, and the tests performed with the output generated will be explained hereafter.

In order to clarify information of the development phase, I will split this chapter into three main parts: generation, sentiment analysis and evaluation of the resulting pieces according to diferent criteria.

Each part will contain the key points of its phase and they will be presented as the natural steps to follow to compose music using the developed techniques.

Although it is split in three apparently independent parts, the development phase was an evolutionary and continuous improvement path and during which it was necessary to go back to previous steps, phases and techniques in order to re-design or re-evaluate previous decisions. Although it was thought to be an autonomous system and concieved as a whole, the parts that formed de development will be taken as independent parts, and treated as independent components according to the techniques used in each one of them.

4.1. Music Theory Concepts

In order to start developing the model, it is worth starting by pointing out some basic music concepts:

Music Notation: the music notation is the system used to visualize music sounds. Music notation is formed by staff, notes, silences and variations.

Staff: the staff is the set of horizontal lines that the notes will be placed on.

Music note: a music note is the symbol denoting a musical sound.

Pitch: is the music frequency of the note.

Duration: determines the length of time for which a note is played.

Music chord: a music chord is a set of notes played in the same time beat.

Beat: is the basic unit of time used in music.

Tempo: is what sets the speed of the beat in the composition.

Scale: is a group of notes arranged in ascending or descending order. The most common scales are the major and minor scales. This scales will be used to develop the theoretical approach to the sentiment analysis section.

Chromatic scale: a chromatic scale is the range of notes by which the music piece will be formed.

Ascending/descending order: the order specifies the succession of notes according to the relation of their pitches. If the pitches go from low to high, the scale is said to be ascendant, otherwise, it is descendant.

4.2. Dataset

In order to train our model it was necessary to define a dataset. A dataset is a set of inputs chosen to train our model.

During the developing of the project, several datasets have been used. The first approach was taken using the Magenta MAESTRO[1] dataset, a dataset composed by piano virtuoso composition.

Moreover, for the testing of the Markov Models, the dataset was enriched with certain classical compositions obtained from Classical Archives[2].

Additionally, due to the musical training already acquired before enrolling this project, I could record myself using an electronic keyboard and throughout the use of Audacity[3], a simple music manipulation and recording software. Using this simple software, I was be able to record myself playing certain pieces on the piano and training the model using those songs.

The main reason why I chose to record myself playing those pieces, and the comparison between the expected and obtained result using them will be explained in the following sections.

4.2.1. Dataset type: the instrument

During the development of the project we will focus on generating music composed and played for piano.

Piano is a widely known instrument, with a chromatic scale big enough to embrace music pieces from different genres, and scales, and therefore we will not limit the resulting pieces to a fixed chromatic scale.

The range of notes stablished in the piano can be seen in the following figure[4]:

4.2.2. Dataset type: notes, chords and songs

As mentioned above, the dataset can be formed by the three main components that music is formed with: notes, chords and songs.

Due to the fact that the developed Markov Models, as it will be explained

[1] https://magenta.tensorflow.org/datasets/maestro
[2] https://www.classicalarchives.com/midi.html
[3] https://www.audacityteam.org/
[4] http://bigit.karikaturize.com/piano-music-notes-chart/piano-key-chart-beginners-here-are-a-few-easy-exercises-to.html

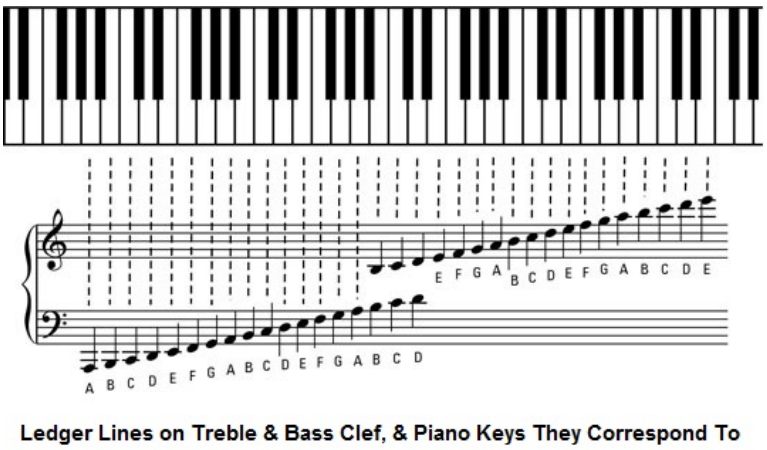


Figure 4.1: Correlation between the chromatic scale and the piano keys

later, is focused on the probabilistic approach of preceding notes, it is necessary to have a full range of notes preceding one-another in order to study the probability of a note to be the succeeding one of a given note.

Moreover, if we look at it from the research approach, as said previously, the aim of this project is not to focus only in music generation, we would also like to generate music based on the learning of already-existing compositions. Therefore, it is necessary to feed the models with the pieces it us supposed to learn from.

4.2.3. Dataset type: Genre

During the development of the work classical music was used to train the models. The reason for it is the variety of notes found in classical music compared to other genres, and, therefore the amplitude on the learning process it impulses. The chromatic variety of the classical genre allows a deeper study of the inputted songs.

4.2.4. Dataset size

The size of the chosen dataset is determined by the number of pieces that forms it, those pieces will serve as the input of the system. It is necessary to take into account the size of the dataset. As, it determines in a big way the behaviour of the model. A dataset too big will make the model to come up with pieces incoherent and formed with a huge amount of information.

The reason why a big dataset, in our case, deploys an unfavourable output is because the model tries to incorporate into the resulting pieces, all the learning acquired during the training phase.

On the other hand, a small dataset, as it was experimented, makes the model come up with a result too similar to the input pieces or note.

These aspects will be explained in the following sections.

4.3. Implemented models

Hereafter, the models created to generate the music pieces, their structure and architecture will be explained. In this module two main techniques will be used: Markov Chains and Neural Networks.

4.3.1. Markov Models

The Markov Model created for this project is nothing else than a matrix where each component (formed by «x», and «y» variables) is composed by the probability that states the chance for a note to precede another note.

If we imagine an usual matrix, the rows and columns that will conform the matrix will be the notes themselves (formed by «pitch» and «duration»), and therefore will be the axis of the resulting matrix. The body, on the other hand, will be the probability of a note to be followed by any other note. In this way, if we read the matrix shown in figure (X), we can say that the note «XX» is followed by the note «YY» with a probability of «ZZ», on the other hand, note «XX» is followed by note «YY» with a higher probability.

The matrix will be constructed from the dataset given to the model (in this case, a song), studying the number of times a given note of the piece is preceded by another specific note, and therefore, its probability.

Once the matrix is constructed by different probabilities of a note to be followed by another note, we will start by randomly picking an element of the matrix. Once an element has been chosen, we will maximize the probabilities creating a tree of one branch, on which, we position one element after the other. The result will be a list of elements by their coordenates (x,y) where «x» is the preceeding note and «y» the successor. The list therefore, if we maintain only the «y» of all elements, and the «x» value of the first element, will be a list of notes that will be the sequence of notes that will form the

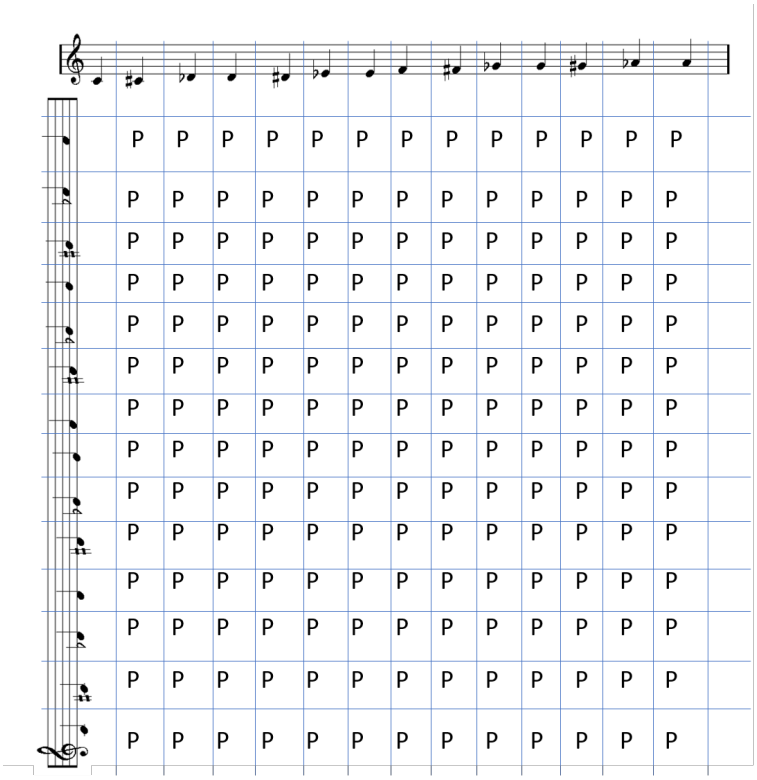


Figure 4.2: Markov Matrix Diagram

final music piece.

In figure 4.2, a high level representation of this matrix has been designed.

The use of Markov Chains is all about predicting the next data on a sequence given some probabilities and maximizing them.

The use of Markov Models for music composition, is based on the piano roll mechanism of implementing music, as it condenses the notes of the musical composition chromatic scale into a matrix taking into account the appearance, and «activating» the cell of the notes the song contains.

4.3.1.1. Architecture of the solution

In order to develop all the above mentioned model, the task will start by dividing the implementation in three submodules:

- Generator Module

- Converter Module
- Markov Module

This structure of the solution allows to see the different steps taken to generate a full musical composition using the Markov Models approach. The internal functioning of each component is explained hereafter:

4.3.1.2. Generator Module

This class will be used as the main driver of the composer method and will be the main controller of the model.

For the implementation of this module, the Mido library has been used for the handling and manipulation of the midi events.

On this class we will create the Markov Chain from the specified song or melody using the Markov Chain Module, and once created, we will generate the new melody that will form the output of the module.

The Markov Matrix will be created using the Markov Chain class that will be explained later. And it will be saved in a structure of a dictionary that will store the different probabilities of the matrices.

Once the Matrix is created, then it is necessary to create a list of notes, extracted from the resulting matrix, that will be converted into midi events so they can form a brand new song. For that purpose, we will use the get next function that will allow us to get, according to the current note, the one that must be the successor of the actual note according to the probability established by the matrix.

The duration of the pieces is determined by the setting of a variable that enunciates the number of note the resulting pieces will have.

4.3.1.3. Converter Module

This module will allow us to convert the midi file into a Markov Chain by extracting the data that the midi contains and creates the list of events that will be treated as the Markov Chain module to create the Matrix.

In order to create de nodes, it goes through the midi file and extracts the duration, pitch and velocity of the notes. From that information, the class will create the sequence of the notes that will form the song in midi format.

The velocity argument of the note is how it known the strength with which the note is played on the composition. This argument will be a part of the sentiment analysis section in the further research section.

4.3.1.4. Markov Module

This last module is the one in charge of creating the Markov Chain from the information parsed by the Converter module and manage this information correctly to create the new generated melody.

This class will create the matrix of notes, formed by duration and pitch of each note that appears on the piece.

The form that this matrix will take it is shown in the following figure.

	64:250	67:250	55:250	48:250	36:250
60	17	9	12	23	5
64	0	2	2	0	0
67	3	4	4	0	1
72	1	8	0	0	1
76	0	2	0	0	0
79	0	4	0	0	0
84	0	0	0	1	2
88	0	0	0	0	0
91	0	0	0	0	0
96	0	0	0	1	0
100	0	0	0	0	0
57	1	0	0	5	0
65	1	0	0	1	0
69	3	2	0	0	0
77	0	0	0	0	0
81	0	0	0	0	0
89	0	0	0	0	0

Figure 4.3: Markov Matrix Output

As we can see in the figure, the columns and rows of the matrix are formed by two arguments: the first argument refers to the midi frequency correlated to the note and its duration. In the following figure we can see a representation:

Additionally to the specified function of the module stated above, some utilities functions have been used:

- **Print Matrix:**

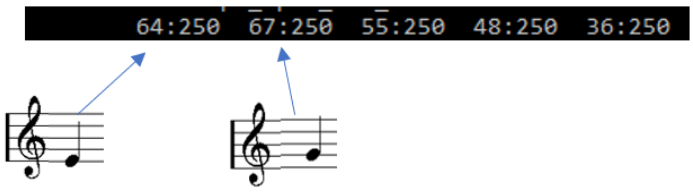


Figure 4.4: Notes staff notation to midi frecuency correlation

This function allows the user to see the matrix built from the input that has been generated. It prints the matrix in the format specified above.It takes as a self-argument a chain of notes extracted from the midi file.

Due to the fact that a big amount of notes may appear in a song, when printing the matrix only a small portion of those notes will be printed, showing the user the most significant part of the matrix, that is, the notes with the highest probability of appearance.

- **Combine two matrices: merge**

  This utility allows joining two matrices of two given music pieces.

  This function allows the model to take two arguments instead of one. The functioning of this version of the model allows to extract the matrices related to each of the arguments (midi files) and combine them to create a new matrix from which to generate the output of the model, as shown in the following figure.

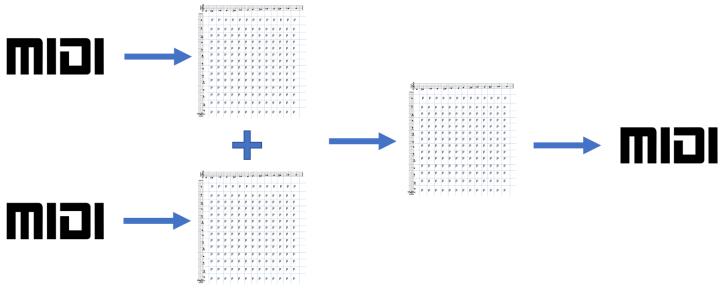


Figure 4.5: Markov Matrix merging

The combination of both matrices is done by summing the values of the same components of the matrix, that is, in our model, if two notes

appear very often in both music pieces, in the resulting pieces will have twice as many probabilities to appear.

This allows us to see the effect of combining two very different musical pieces. If, using this functionality, we join a Wagner composition and a Mozart composition (two composers with very different chromatic scales in their composition), the resulting piece will be one that contains influences of both composers.

Using this function, it allows us to enrich the resulting composition with different composers and styles.

Given the fact, that the Markov model also takes into account the tempo (duration) of each note, by combining two pieces from different genres, it will allow us to find intermediate subgenres between the two inputs given to the model. This functionality was tested with combining Jazz music with Classical compositions.

Therefore, the full model behaves according to the following diagram:

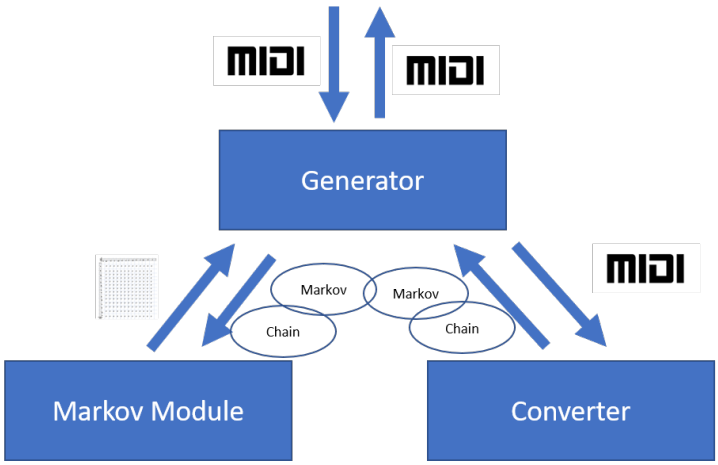


Figure 4.6: Full Diagram Flow

4.3.1.5. Model execution

All the above mentioned model can be excuted in a python environment throughout the CMD command prompt.

As specified in the architecture section, the generator module will be used as the main driver of the program, therefore it will only be necessary to run this specific program.

It is needed for the program to run correctly to set pass several parameters:

- **Input Songs**

  This parameter specifies the song that will be used to create the resulting matrix from which the model will generate the output of the program

  It is worth recalling that an extra song can be inputted. The model will create the Markov Matrix associated to each of the songs, and then, using the merge function, will merge both matrices.

  Then, the model will create the new song from the resulting matrices.

  The model it is prepared to automatically detect which of the two version (with or without merging) the user has chosen, taking into account the number of arguments the program is given.

- **Output song**

  This argument specifies the name of the output midi file where the generated song will be saved.

- **Number of notes**

  Additionally, the user is also capable of specifying the number of notes that the resulting song will have, and therefore, the duration of the resulting song is determined.

When executing the model, the program will print the most significant part of the resulting matrix of both, input and output of the program.

The following figures show how the program is executed with the above mentioned options taking as input Chopin pieces:

4.3.1.6. Output handling and evaluation

The output of the system consist on a midi file, containing the composed music. The output result inherits the music values established in the original input file, such as tempo, clef, etc.

```
>generator.py chp_op18_format0.mid out.mid
        70:250  46:250  74:250  68:250  65:250
70      97      34      13      81      38
46      53      30      11      17      31
68      105     7       14      144     49
74      11      4       7       7       3
58      117     35      12      59      45
65      44      10      9       39      89
51      33      9       10      65      15
67      52      22      7       26      31
75      43      4       4       45      27
63      69      22      7       69      32
50      8       4       4       11      14
61      69      11      4       41      29
```

Figure 4.7: Execution of the piece with Chopin Opera 18

```
>generator.py chp_op18_format0.mid chpn_op7_1_format0.mid out2.mid
        70:250  46:250  74:250  68:250  65:250
70      104     42      33      81      40
46      59      36      11      17      31
68      105     7       14      144     49
74      11      17      14      7       3
58      118     67      28      59      47
65      44      10      9       39      95
51      33      9       10      65      15
67      58      34      7       26      37
75      43      8       22      45      27
63      141     64      27      69      32
50      8       4       16      11      14
61      69      11      4       41      29
```

Figure 4.8: Execution of merged pieces with Chopin Opera 18 and Opera 7

Moreover, to complete the execution of the program, a utility called Lilypond library to create music «.ly» objects, capable to be converted to music sheet using MuseScore.

The use of this utilities allowed to come up with the music sheet of the resulting composition.

Moreover, considering this project is embedded in a bigger project that aims to create Musical Poems, it allows us to merge music and voices in the same music sheet using MuseScore as the main tool to do it.

4.3.1.7. Model testing and adjusting

During the development phase, several tests have been performed to check the correct functioning of the model. The tests performed have included:

- **Testing over repetitive notes:**

  In order to test the program, making use of Audacity, several piano songs have been recorded. Those songs contain only one note repetitive over and over again.

  As expected, the resulting matrix of the song shows that, the probability is condensed in one node of the matrix, and therefore, the resulting song can only contain the same note it was given as input.

  This test allowed to check the correct functioning of the matrix probability calculation as well as the musical repetition patern.

- **Testing with ascendent notes:**

  Following the above mentioned example, a recording of an ascending pattern of three notes have been recorded and served as input of the model.

- **Repetitive testing over the same song:**

  During the early stages of the model, and concretely during the first phases of testing, it was observed that the resulting generated songs did not vary if the inputted song was maintained.

  The reason for it is because the first approach to pick the first note of the song, from the previously built matrix, was done by picking the highest probability, and the successors were determined in decreasing order of that probability.

  To solve this issue, the model was corrected incorporating a clarification on the Markov Chain Module. The module will randomly pick a note from the matrix and determine the successor in descending order.

  The random picking of the first note makes the model create a different song each time, regardless if the input song varies or not.

4.3.2. Neural Networks

As Markov Models only generate a subset of the given data and as result, the generated pieces may sound very familiar to the ones used to generate the matrix if the piece is too small, the next natural step was to investigate the possibility to create brand new composed music pieces.

Neural Networks will allow us to perform the following task: generating music from a previously learned «training». The way neural networks learn, it is very similar to the learning process of a person. The person will learn music by listening and reading music and, if any person is asked to compose music, the resulting piece will have influeces from the pieces that person learned from.

The Neural Network will use this principle, and will learn to compose music starting from the input that is given. Therefore, the next note generated by the net, will be determined by the dataset with which it was fed. This fact causes the net to create music with no time regulations or adjustments, that is, the resulting music piece will not have a soft part preceding a hard or more aggressive part. This is due to the nature of the Network, as we are talking about the most simple version of a Neural Network, it is calculating probabilities based on independent cases and its aim is to be able to match and give the best option for any event that might happen along the play, regardless of structure, meaning, or other variables related to long-term previous learning, as the input is independent.

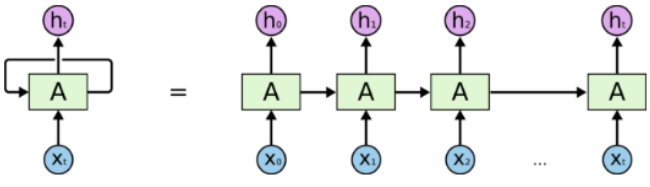


Figure 4.9: Diagram of an unrolled neural network

To improve this fact, it was necessary to take into account a type of Networks called Recurrent Neural Networks (RNN), that allows us to take into account previous training. This Network allows to predict outputs based on previous input as well as the current one. But this solution is not the definite one, as, due to the nature of the network and what is called the

"Descendent or Vanishing Gradient Problem". That problem is that, as more layers using certain activation functions, the gradients of the loss functions approaches to zero, making the network hard to train. To give a simple explanation, as the layers using the same activation function grows, the computational effort, time also grows and the result becomes incoherent and noisy.

The result of this problem is consequence of increasing time in the network computation training time as well as not being able to ensure the correct prediction for the network as well as the lack of depth in the model created. The depth in the model increases the capability of the performance of the network but, at the same time, increases the computational effort and time needed.

The solution for this problem was the use of LSTM Networks which allowed to create music pieces with time lapses changes along the play.

Their capabilities and their flexible design and decision making approach allows us to model and create customizable schemes for our music composing system.

To determine the datasets used, we picked the datasets created by Magenta to compose our music given due to the variety it offers as well as the richness of their datasets.

We focused mainly on compositions for piano to keep it simple and to make the composition and the evaluation of the resulting pieces more handleable. We also consider piano to be the most versatile instrument in terms of music composition as it has a great range and it fits in nearly every music genre.

4.3.3. Architecture of the solution

In order to create the final solution regarding the music composition, a previous design was developed and planned. This solution took into account the music composing methodology that ran inside the generator as well as the communications among the other parts of the project.

The music composer module, formed by two submodules explained in the following point (Markov Models and LSTM Networks), will interact with the evaluator component in the following way.

The Music composer creates music according to the available methods,

and the outputs, a MIDI file and a music sheet, will be passed to the evaluator, who, after performing the corresponding calculations and methods, will give the components feedback in the following form:

- Changes on a certain note or group of notes: the evaluator can identify which notes do note match with the given music and the Lyrics generated and tells the composer which notes to be changed.

  The composer has a module on modifications with utilities functions in order to perform available modifications.

- Make music sadder or happier according to the sentiment analysis performed.

  In this case, the music composer can make use of the Markov Chain modules to generate the resulting Markov Matrix of the composed piece.

  Then, it will calculate de Markov Matrix of a sadder or happier song, depending on the requirements and will merge those two matrices. The happier or sadder song will be picked from the corresponding dataset.

  After the new matrix has been generated, a new song will be created and the result, as the "parets" are sad, it will be sad. During the development phase will see if, this process has to be performed several times in order to appreciate the change in the music composition. In case several runs are needed, we will straight away apply as many as needed for it to take effect.

4.3.4. Neural Networks

To keep exploring the possibilities of Computer Generated Music the next approach taken was to create a simple model based on general Neural Networks using Keras. The model will allow,to test the different Neural Netwoks models (such as LSTM), as well as setting the parameters of the networks (activation functions) and defining the created model.

The use of RNN will allow two main things:

- **Creation of music by the iterations over a Neural Network:**

As we talk about a RNN, the network is fed with recurrent data. That means that the data that enters into the net, is combined with the data generated by previous steps in the training phase.

As we create the model using LSTM, the resulting output will take into account all input regardless of how back in time they are.

- **Get rid of the vanished gradient problem and therefore create richer compositions and better training.**

  The Vanished Gradient Problem states that, when training a Neural Network, due to the nature of the training, the network will try to cover all possible outputs, and take into consideration all variants along the song and could cause the vanishing of part of the information along the way.

  That fact drives us to conclude that the resulting tests performed with RNN no LSTM, did not provide the pieces with a defined structure, variants or modulation, and as we will see, create pieces filled with eighth notes.

  Due to the nature of the LSTM network, the resulting song should have a defined structure and not be monotonic or seem too random.

4.3.4.1. Network type

The approach taken towards Music Generation using Neural Netowrks, focused in the developing of a Recurrent Neural Networ, and the use of a Long-Short Term Memory Network (LSTM).

The model was created using using Python Anaconda Spider as the main tool to test and implement the model.

4.3.4.2. Music parsing

We started by creating the formatting input of the LSTM network as a sequence of events using Music21. Given that MIDI files is coded this way, the resulting parser is a simple implementation that saves the midi events into an array.

Due to the fact that RNN behave better with numerical values, rather than music notation values such as ABC notation, it was necessary to create

a parsing function to create transpose the different inputs into a numeric value.

Therefore, the resulting input of the network will not be a midi file, but a set of numerical consecutive values.

4.3.4.3. Input trimming

In order to decide the amount of information given to the network as the input parameter, as well as the information obtained from it, it is important to «trim» the data into smaller pieces.

4.3.4.4. Data normalization

It is not strictly necessary to normalize the data, but it has been shown that the AI algorithms perform better with normalize data when dealing with big datasets and eases the back propagation training process of the networks (Sola and Sevilla, 1997).

For that purpose, we started by parsing the midi file from the input and turning it into a series or sequences of event formatted in order to become the input for the Neural Network. The format chosen is to define a correlation list of lists of notes an integer values.

4.3.4.5. Model Creation

The next step is to create the model itself. The model will be formed by a Neural Network formed by several training layers.

We will also include some regularization layers with Dropout Srivastava (2014). Dropout is a technique for regularization of neural networks that allows the dropout of random neurons along the network to avoid over fitting. See figure 4.11.

In the model definition, we will also include the activation function chosen. During the development of the project, several activation functions have been tested. The results obtained form the tests will be explained later on in the chapter devoted to analyse the obtained results.

The activation functions (see figure 2.2) tested are:

- **Sigmoid**

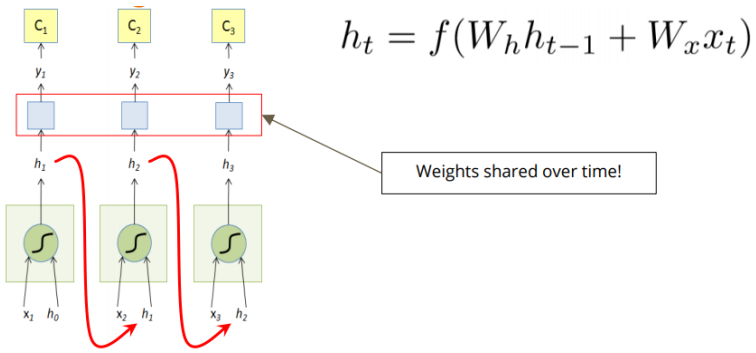


Figure 4.10: Dropout layers and overfitting

The sigmoid activation function is a function whose values range between 0 to 1 that has the shape of an S

- **Hyperbolic Tangent**

  It is defined between 1 and -1

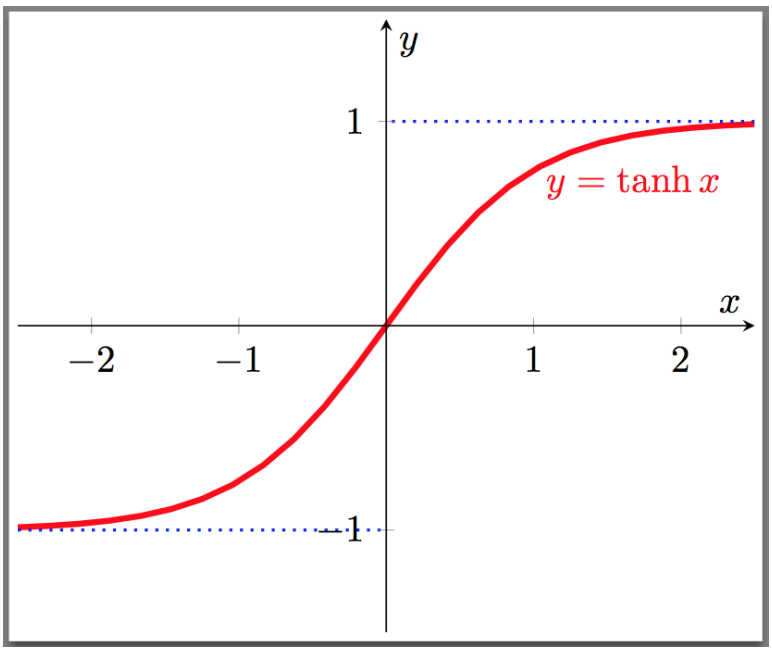


Figure 4.11: Hyperbolic tangent graph

- **Rectified Linear Unit (RELU)**

  This activation function comes to resolve a problem attached to the use of the Sigmoid and Tangent functions called the Gradient Problem.

Apart from fixing the gradient problem, it also provides more sensitivity to the output of the nodes and avoids saturation of the nodes. The use of this function also provides the possibility to decrease the number of dropout layers in the neural network.

The function works in a similar way to the max function in common mathematics. It return the maximum of two given values, but it has the characteristic that half of the input for the RELU function will deliver a lower boundary value previously set (normally zero), and at some point in the development of the function, will start performing like a linear function.

This fact, makes the activation function to filter the output and discard or delete the union of two given nodes. We can imagine, the activation function works like a switch, deactivating or activating the connections of the nodes depending on the inputted value.

Hereafter, we will show some advantages we could see using this function as the activation function, some of them recalled in (Glorot et al., 2011):

Computational Simplicity

As it can be coded as a max function.

Representational Sparsity

This function is capable of returning a zero value, unlike the others that will always return an approximate to zero value

Linear behaviour

As it behaves very similarly to the linear function, but the fact of setting the lower boundary, allows the function to be more sensitive and returning a more filtered output.

Used in Deep Neural Network

Due to the Rectified Linear activation, and the back propagation of the recurrent networks, it allows fo the networks where it is used to be trained faster and with better results in computational terms, and as it was shown during the testing phase, better musical results.

Moreover, Keras allowes us to change some characteristics of the model implementing different optimizers as well as different activation functions.

The optimizer is a parameter for the Keras Neural Network model created that has the aim of optimizing the performance of the Network and checking the effects of activation function modification in the resulting music pieces.

There are several Optimizer we have tested:

- **Adelta** (Zeiler, 2012)
- **Adam** (Kingma and Ba, 2014)
- **Adamx** (Kingma and Ba, 2014)
- **Nadam** (Dozat, 2016)

Although they all have been tested, the one that delivered the best result was the RMSprop, as said in (Bengio, 2015).

The above mentioned parameters were set and adjusted during the testing and developing phase and the information about them has been collected from the Keras Documentation[5] and from the cited sources.

4.3.4.6. Network Training

In order to train the model, the previously prepared data will serve as input.

To train the model, it is necessary to specify the number of times the model will go through the training data set, that is, the epochs or iterations, as well as the number of items that will form the training dataset.

After the training of the network has completed, the resulting weights are saved into a «.hdf5» file.

4.3.4.7. Music Generation

Finally, in order to generate the resulting piece, it is necessary to combine the information collected throughout the training, that is, the weights, and input data.

It is worth recalling that the input data, as well as the training data, have to go throughout the same parsing. This implies that we have to convert the input data into the sequences of numerical values, as the training data in order to be input to the model as it has been previously explained.

[5] https://keras.io/

To generate the array of notes, it is necessary to specify the number of items, that is, notes, and this allow us to define the length of the resulting piece.

The output of the model, therefore, will be an array of notes.

4.3.4.8. Output handling: reverse parsing

In order to convert the array of notes and chords into a music piece, it is necessary to decode it, that is, perform the opposite process performed in the Data Preparation Section.

The list of notes, then, will be converted into a Music21 Object and, with that item, we can save the corresponding Midi file, finally obtaining the result of the model.

The resulting midi file will can be subjected to the same process as the output from the Markov model in order to convert it into a Music Sheet using Lilypond.

4.3.5. A theoretical approach to Sentiment Analysis

This part of the project states a theoretical approach to Sentiment Analysis, proven and tested with the models specified above.

In order to create music, we must not forget about the sentimental part. The Sentiment Analysis is the science or discipline that studies the facts and ingredients that are necessary to create moving music pieces or compositions.

Driven by the possibilities of musical creation that the chosen methods provided us, it was considered the best option to create a personal and own strategy for music composition to create sentiment analysed music.

Given the fact that this project is thought to be included in the project presented by Marisa y Carlos, the strategy must be as simple as possible in order for the evaluator to give feedback about the "score" of music that will be analysed.

Given the fact that the factors or variables that the evaluator will take into account are directly related to style and music theoretical approach inherited from pieces that were the originators pieces, that is, the output will inherit some factors from their "parents" that the evaluator will take into account.

Bearing that in mind, we will have to take them into account in order to

regenerate music, several factors such as those ones are metronome (tempo), key, mode, etc.

Therefore the main idea is to create a personalized dataset that contains classified pieces of music and according to the feedback given by the evaluator, the re-generation or modification of music, will take into account two main facts: the original dataset from which the song has been generated and the creation of a new dataset to regenerate the whole song or part of it to comply with the given feedback.

The resulting piece, therefore, could be generated by a Wagner Dataset, but if the Evaluator module created by Carlos, tells us to soften a the musical composition, we may introduce into the dataset some Chopin sonatas as they are characterized for being in a slow tempo and lower pitch of the chromatic scale, compared to the Wagner composition.

4.3.5.1. Tempo & Scale

To figure out how this datasets can be created we had to stablish some initial constraints or assumptions and if the feedback of the song spins around the sadness or happiness of it. We take into account this two factors: tempo and scale.

- **Tempo of the song:**

  We will assume, as stated in (Wassermann et al., 2003), that most of the slow music compositions created in classical music, tend to be quiet and calm or sad. Therefore, the songs created or composed from pieces with this ingredient, are more likely to be sad or calm.

- **Scale of the song**

  If the scale of the song is composed by major chords or notes that are normally high, we assume the song is a happy one and therefore the resulting pieces generated from this piece are more likely to be happier.

4.3.5.2. Velocity

As told before, velocity is an argument in a midi event that is used to determine the strength with which the note is played by the instrument.

As we can all recognize, aggressive and happy songs, are related more to notes being played with a high intensity, which in music would correspond

to forte or mezzoforte modulations. The main aim of this fact is that the pieces try to transmit a sense of joy and pleasure to the listener.

On the other hand, sad or melancholic songs are normally related to a softer play, a more delicate and probably quiet sound.

Bearing this in mind, therefore, by the analysis of the music pieces, we can recall that the quiet songs will be more likely related to sad songs while "harder" songs will, somehow be related to happy songs.

The sentiment analysis module will take all this into account and will drive the instructions from which the music will be changed and modified according to the feedback given by the evaluator.

Even though the sentiment analysis module is based on a simple principle, as shown in the testing phase, returns good results and the outputs can be considered sentiment sensible. Of course, it has some limitations, due to the size of the data set and the nature of the methodology, but as the approach returns good results, it can be considered a success.

4.3.6. Learned lessons and result comparison

Hereafter I will point out the most significant keypoints concerning outputs and models creation:

- **Importance of Parameters**

  In both models the values set to the parameters that control de model, it turned out to be highly important, and conditions the results obtained throughout the use of the model.

- **MIDI manipulation**

  Regarding MIDI and audio manipulation has been surprising the amount of possibilities to handle and manipulate those files

  To handle music objects, several libraries and sources were found and many of them, tested. We focused on two main modules, mido and music21 as MIDI handlers, but other, such as pretty-midi were tested.

- **Approach of Piano Rolls**

  During the first few steps on the creation of the Neural Network model, the possibility of using piano rolls as the input was conceived. A piano Roll was a structure used a the begining of music history as a roll

perforated paper which controls the movement of the keys in a pianola or similar instrument.

This approach was attractive, because, as a computer structure, the piano roll is nothing more than a matrix or a dictionary. As the approach seemed attractive we decided to start exploring this path and we starting creating the system to handle piano rolls and convert them into handleable music objects.

- **Notation**

  On the notation field, we also tried different approaches, such as ABC notation. The use of ABC notation to fill the models were discarded due to the previously explained fact concerning the Network Capabilities and its decrease when using no numerical values.

  ABC notation was attractive at the beginning due to the simplicity and the familiarity of the notation. The notation allowed simple CSV modifications to be performed as well as many other operations.

As can be seen, many paths leaded to empty roads during the developing of the Music Composers, but from all paths we learned something new, and we do not discard to use them later on in the future work.

All approaches taken were valid, but we only kept the ones we thought, had possibilities to serve our aim.

4.4. Results and conclusions of the developing phase

During the testing phase of the developed models, some problems were faced and some conclusions were deducted from the results.

We can recall the next conclusions depending on the model being tested:

- **Neural Networks**

  Composition Time:

  Given the complexity of the process in computational terms, the Neural Network returned good results when trained for a long time.

  Even though the training time highly improved the result, not much difference was achieved when the time overpassed six or seven hours.

The improvement, on the other hand, was very palpable when comparing the result of just few minutes or even an hour.

This process allowed to see that in some of the cases, not a better result was retrieved by increasing the number layers of the neural network but by the number of epochs.

Rich results

The outputs of the net is highly improvable in music terms but, is still audible and could be considered a rich results due to the slight changes in tempo, measure and pitch that the pieces offer.

We would like to recall that the assumption made at the beginning of the project stating that the LSTM Network Strategy would give better results than Simple Feedforward Network, the assumption was proven true in one quick experiment performed.

In this experiment, the architecture of the model and the layers arrangements were changed. The new arranged model consisted in a Feed Forward Network. Due to the use of the feed forward networks, the computational time of the model decreases significantly, allowing to deepen the Network structure.

The results of this experiment was that, even though the computational effort was lower, the complexity inner functioning of the model was also simpler, and the dataset was maintained, the testing results outputted a bad result in musical terms, confirming the initial hypothesis of the project of the advantage obtained when using LSTM rather than other models.

Moreover, if we compare the results of the first functioning models tested, the eighth notes problem disappear and the resulting composition is note variety enriched.

Method with huge improvement posibilities

As we did not focus on one specific genre or style, this model can be used to compose any kind of music with different results. The model can be changed or personalized to the specific genre of the music that can be created by applying constraints on the model or changing the dataset.

Variety and constraints

Even thought due to the nature of the system, the Neural Network is not constrained, it offers a huge possibility of improvement if some constraints are performed.

In order to develop those constraints, we can focus on the music type, genre, or the output we want to get.

Development tediousness

The developing phase of the neural network was probably one of the most tedious part of the project. Not only because the difficulty that brings attached the fact of creating music with a Neural Network and the different phases of testing that it carries and their corresponding computational time.

The testing phase included the change of the parameters of the neural networks, the neurons layers, the dropout layers, the increase of the epochs time, etc.

One of the most important factors of the difficulty that carries is the amount of time that it takes to train the Network in order for it to generate music.

Some conclusions on this matter will be:

- Increasing the number of layers of the model, that is, its depth, improves the result obtained, but it also increases the computational time and effort for the model to be trained.
- On a certain point, a threshold is reached and the learning capabilities and the result of the model becomes linear, which implies that, augmenting the number of epochs (iterations) over the dataset performed by the model. Once that threshold is reached, a similar phenomena to over fitting appears, which implies that the results of the Networks become noisy as the early stages of the training.

- **Markov Models**

Computation time

In the case of the Markov Model, the time to achieve an audible composition was shorter, but the result was more similar to the one given

by the original piece. In the resulting composition, the structure of the piece was kept and the coherence of notes and the rhythm of it.

Richness results

In the Markov models, as we found out that the resulting pieces was similar, we started combining markov chains to create a brand new Matrix in order to compose a piece that stood apart from their parents. This approach also allowed us to play with the sentiment analysis factor and enrich the composition.

Chapter 5

Conclusiones y Trabajo Futuro

El trabajo desarrollado durante los meses de investigación e implementación me han permitido evidenciar las enormes posibilidades de la creación de música a través de técnicas compositivas derivadas de la Inteligencia Artificial.

Las técnicas utilizadas y las diferentes implementaciones desarrolladas y testadas de dichas técnicas, las cuales, como se han expuesto anteriormente, conllevan ciertas ventajas y desventajas propias, me han permitido ver las posibles aplicaciones de dichas técnicas a partir de la explotación de las ventajas que cada una de ellas conllevan.

El proceso de investigación, aunque laborioso debido a que las técnicas de desarrollo y explotación de la AI están en un proceso de crecimiento y mejora continua, ha resultado ser tremendamente gratificante por las posibilidades de aprendizaje que estas ofertan.

Las composiciones resultantes del uso de los modelos, aunque no resultan ser perfectas en términos musicales, son una muestra palpable de las posibilidades que dichas téncias ofrecen, así como de los posibles usos futuros, más allá de la mera generación musical, que sean de aplicación a la vida cotidiana, o doten de posibilidades a otras áreas.

5.1. Composición para improvisación teatral

Una de las mejoras que me gustaría que fuera implementada en un futuro, es una composición a partir de la conversación de dos personas, por ejemplo

en un entorno de teatro de improvisación. En los últimos años, el crecimiento de las obras de teatro improvisado han aumentados significativamente, y, en dichas obras, el uso de música se ve sujeto al discurrir de la interpretación, que, al tratarse de una obra de improvisación, es libre e impredecible.

Mediante un modelo el cual "escuchase" la obra de teatro, y el tema principal sobre el cual los actores interpretarían, sería útil generar una banda sonora que, a partir de la conversación y el tema tratado, el sistema fuera capaz de modular el tempo, el volumen o las dinámicas de la composición acorde al discurrir de la interpretación de los actores.

5.2. Terminación de obras clásicas

Otra posibilidad de uso de estos modelos, es la "terminación" de las obras de grandes compositores, que nunca se llegaron a acabar. Es decir, que el modelo, a partir del estudio del las obras compuestas del músico, sea capaz, dada una parte de una obra sin terminar, generase lo que sería la continuación de dichas obras. Esta utilidad llevaría consigo un estudio comparativo entre las adaptaciones que músicos posteriores a dichos compositores hicieron de las obras inacabadas, y de las obras que el modelo es capaz de generar.

Esta técnicas sería aplicable a obras como el Requiem de Mozart en su pieza "La Lacrymosa", la cual, el propio Mozart no pudo terminar. Este también sería el caso de la Sinfonía número 8, la conocida como la "Sinfonía Inacabada" de Schubert quien, como indica el alias acuñado, tampoco pudo acabar dicha obra, o la Novena Sinfonía de Bruckner, también inacabada.

Esta idea, de completar obras clásicas inacabadas a partir de técnicas compositivas computacionales, se ha implementado ya por grandes compañías durante el desarrollo de este proyecto. Éste es el caso de la Sinfonía Inacabada de Schubert, que fue terminada por un sistema de AI diseñado por Huawei (Huawei, 2019) con un resultado que los críticos musicales tintan de artificial y carente de alma (Mantilla, 2019).

Aunque la crítica musical no haya recibido de la manera esperada este tipo de iniciativa, desde el punto de vista investigador y desarrollo de cara al estudio de las capacidades computacionales en términos compositivos resulta de gran interés este tipo de estudios y actividades ya que de ellos se extraen lecciones aprendidas de gran valor que pueden impulsar la generación de música como campo de investigación a un nivel superior y poder, de manera

adecuada, utilizar este tipo de iniciativas para el estudio y mejora de los sistemas compositivos modernos.

5.3. Composición para películas y videos

Tal y como se ha explicado en la sección de Estado del Arte, el compositor Alemán Hans Zimmer, ha decidido invertir y apostar por la composición musical a partir de téncicas generadas computacionalmente con su contribución a la plataforma VJAM.

Estas técnicas, podrían a partir del estudio de los colores, las formas, la velocidad, y otros parámetros establecidos componer música para películas y videos. Esta técnica también sería utilizada, por ejemplo para dibujos animados y otros medios de visualización.

Durante el desarrollo de este proyecto, se ha estudiado el caso de Muphic (Johan Bertrand, 2012), un sistema de composición musical basado en imágenes estáticas a partir de las cuales se compone una banda sonora para dichas imágenes.

5.4. Integración con asistentes de voz

Durante el desarrollo del proyecto se ha planteado y se ha estudiado la viabilidad de implementar un sistema de composición que fuese capaz de integrarse con asistentes de voz como Alexa y Google Home.

El uso de dichos asistentes ha crecido exponencialmente en los últimos años, y resulta un campo interesante de investigación de cara al desarrollo de skills para dichos sistemas. Éstos skills serían capaces, por ejemplo, de componer dinámicamente canciones de cuna para niños pequeños, o música de ambiente para incentivar la concentración, explotando teorías como el "Efecto Mozart" (Jenkins, 2001), el cual establece que la escucha continuada de piezas musicales compuestas por Mozart, dada su naturaleza y dinámica compositiva, se mejora en el largo plazo la concentración las capacidades de desempeño.

A partir de los estudios reiterados de las obras de Mozart, la skill sería capaz de generar música basada en las tendencias usadas por Mozart en sus obras y, de tal manera, incentivar el llamado "Efecto Mozart" en aquellos que la escuchasen.

5.5. Aplicaciones en el ámbito educativo

Tanto este proyecto, como las técnicas utilizadas y los resultados obtenidos resultan relevantes para los entornos educativos de cara a enseñar música a estudiantes de diferentes edades.

Dado que la generación de música mediante los métodos utilizados explotan los factores creativos vinculados a la música, al mismo tiempo que incentivan la creatividad y la curiosidad por los entornos de desarrollo de software y tecnologías emergentes.

Como se puede observar, mediante las líneas definidas como trabajo futuro, el espectro de aplicación de este sistema compositivo, así como las técnicas utilizadas, dejan abierto un amplio abanico de posibilidades para la creación de nuevos sistemas y aplicaciones en diferentes ámbitos, contribuyendo así a la divulgación de las técnicas generadoras y teoría y generación musical.

Chapter 5

Conclusions and Future Work

The work developed during the months of investigation and implementation of the models allowed me to evidence the great possibilities that the music generation field has to offer, as well as digging into the developed AI-related techniques.

The used techniques and the different implementations developed and tested, which, as already said, have attached their own already-specified advantages and disadvantages, allowed me to evidence the possible applications that those techniques offer, bearing in mind their nature.

The research phase of the project, although sometimes difficult due to the early stages on which the techniques are, and the amount of information related, allowed me to see the growing process and continuous improvements these techniques are subjected to. These facts, as well as the continuous learning and improvement process have been greatly satisfying.

The music composition generated throughout the execution of the models, even though can not be considered to be perfect in musical terms, can be considered as a tangible proof of the possibilities that the Markov Models and Neural Networks have to offer.

In this final chapter, I will expose the main ideas and uses these techniques can be subjected to, beyond the mere music generation process.

5.1. Composition for theatre improvisation

One of the use and improvements that I would like to see in the following years to come, is the use of these techniques when creating the soundtrack of a conversation had by two people. In the last few years, there has been an increasing tendency of the production of theatrical productions based on the improvisation of two or more people. In those production, the use of music is subjected to the flow of the conversation, and by nature unpredictable and free.

The model must be able to "listen" the show and improvise, along the actors, possible music accompaniments that fits the subject brought to action. This composition must be able to change along the way throughout the developing of the play, and adjust music parameters such as dynamics, to fit the play.

5.2. Classical pieces completion

Another possibility that arises from the use of these models, is the completion of classical music pieces that, due to different factors, never got to be finished. This implies that, the model, starting from the studying of classical compositions, must be able to, given a part of an uncompleted song, complete the given piece, following the tendencies and characteristics of the original composer.

This technique would be aplicable on pieces such as the Mozart Requiem, in the Aria for choir «Lacrymosa», which, he was not be able to finish. Additionally, this technique could also be applicable to the Schubert's Eighth Symphony, popularly known as the «Unfinished» or the Bruckner's «Nine Symphony», which, as «Lacrymosa» were not finished by the original composer.

Big tech and technological companies have be focused on the developing and arising tendency of these techniques. This is the case of Huawei (Huawei, 2019), which tried to finish up the «Unfinished Schubert's Symphony». Even though the results have not been the best ones expected (Mantilla, 2019). , the fact that companies such as Huawei interest themselves on the developing of these techniques, certainly shows that they are facing the right way and it is worthy to keep researching.

5.3. Composición para películas y videos

As it has been explained in the section devoted to the «State of the Art», the german composer Hans Zimmer, decided to invest and get on board on a project focused on music generation from computer generated techniques throughout the VJAM platform.

This techniques, as Hans Zimmer, could be focused on the creation for music for films, videos and cartoons.

In this same research line, I would like to recall the studied brought up in Muphic (Johan Bertrand, 2012), a composition system based on static images from which a music is composed that matches the images.

5.4. Voice assistant integration

During the development of the project there has been an unsuccessful approach to the viability to include the models into a skill created for voice assistants such as Google Home and Alexa. This skill would be able to compose music using these assistants.

The increase in the use of these assistants have grown exponentially in the last few years, and they turned out to be an interesting field for the development of skills to enrich those systems. These skill would be capable, for instance, of dynamically composing music pieces such as lullabies or ambient music to encourage concentration. This creations, would, at the same time, exploit the "Mozart Effect"(Jenkins, 2001), which states that the continuous hearing of musical pieces composed by Mozart, given their nature and dynamic composition, causes improvements in the long terms for people's development capacities.

From the reiterated studied of Mozart pieces, the skill will be able to generate music based on trendings used by Mozart in his compositions and, therefore, impulse the, so called, "Mozart Effect" in the listeners.

5.5. Applications in the educational environment

This project, as well as the used techniques to develop it, and the obtained results, turned out to be relevant for educational environments in order to teach music to students of all ages.

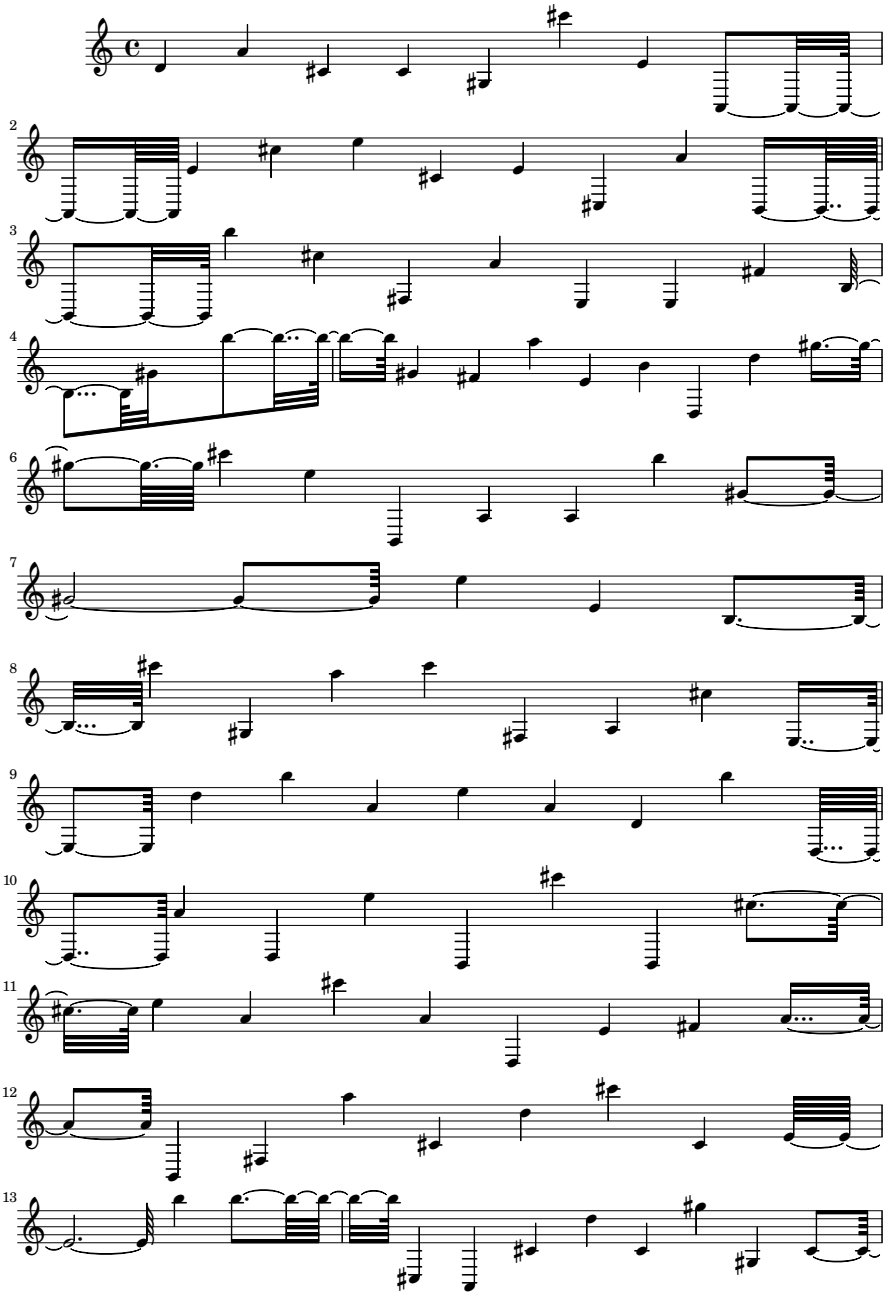
Given the fact that music generation throughout used methods exploit and try to imitate the creativity factor inherent in human beings, at the same time, they expect to impulse the use of emerging technologies and software development activities.

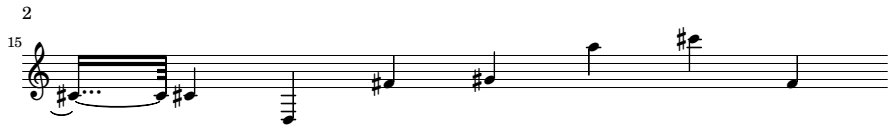
As it has been proven, throughout the different activity lines defined for future word, the spectrum conceived for the application of music generation systems, as well as the techniques used to develop them, leave open a huge range of possibilities for the creation of of new systems and applications in different fields, including popularization of generation and composition musical theory and techniques.

Annex A

5.6. Annex A: Output Sample

The following music sheet is a sample of the outputs generated by the system





Bibliography

If I have seen further than others, it is
by standing upon the shoulders of giants.

Isaac Newton

Antonio Caparrini López, L. P. M. Genre classification of electronic music. 2017.

Aucouturier, J.-J. and Pachet, F. Representing musical genre: A state of the art. *Journal of New Music Research*, Vol. 32(1), 83–93, 2003.

Basharin, G. P., Langville, A. N. and Naumov, V. A. The life and work of aa markov. *Linear algebra and its applications*, Vol. 386, 3–26, 2004.

Bengio, Y. Rmsprop and equilibrated adaptive learning rates for nonconvex optimization. *corr abs/1502.04390*, 2015.

Boylan, J. H. Hal in" 2001: A space odyssey": The lover sings his song. *Journal of Popular Culture*, Vol. 18(4), 53, 1985.

Braga, M. The verbasizer was david bowie's 1995 lyric-writing mac app. 2016.

Bryner, B. *The piano roll: a valuable recording medium of the twentieth century*. PhD thesis, Department of Music, University of Utah, 2002.

Carr, C. and Zukowski, Z. Generating albums with samplernn to imitate metal, rock, and punk bands. In *Proceedings of the 6th International Workshop on Musical Metacreation (MUME 2018)*. 2018.

Copeland, B. J. and Long, J. Turing and the history of computer music. In *Philosophical Explorations of the Legacy of Alan Turing*, 189–218. Springer, 2017.

Dozat, T. Incorporating nesterov momentum into adam. 2016.

Ellis, D. P. Extracting information from music audio. *Communications of the ACM*, Vol. 49(8), 32–37, 2006.

Glorot, X., Bordes, A. and Bengio, Y. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 315–323. 2011.

Gouyon, F., Dixon, S., Pampalk, E. and Widmer, G. Evaluating rhythmic descriptors for musical genre classification. In *Proceedings of the AES 25th International Conference*, 196–204. 2004.

Horner, A. and Goldberg, D. E. Genetic algorithms and computer-assisted music composition. In *ICMC*, Vol. 91, 479–482. 1991.

Huawei. HUAWEI PRESENTS UNFINISHED SYMPHONY. `https://consumer.huawei.com/uk/campaign/unfinishedsymphony/`, 2019.

Igoudin, A. *Impact of MIDI on electroacoustic art music*. CCRMA, Department of Music, Stanford University, 1997.

Jenkins, J. S. The mozart effect. *Journal of the royal society of medicine*, Vol. 94(4), 170–172, 2001.

Johan Bertrand, V. G., Carlos Catalán. Muphic: Composición musical automática basada en imágenes. 2012.

Karpathy, A. The unreasonable effectiveness of recurrent neural networks. *Andrej Karpathy blog*, Vol. 21, 2015.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Knees, P., Pohle, T., Schedl, M. and Widmer, G. A music search engine built upon audio-based and web-based similarity measures. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 447–454. 2007.

Kouemou, G. L. and Dymarski, D. P. History and theoretical basics of hidden markov models. *Hidden Markov Models, Theory and Applications*, Vol. 1, 2011.

Mantilla, J. R. Un algoritmo completa la misteriosa 'sinfonía inacabada' de schubert. 2019.

O'Shea, K. and Nash, R. An introduction to convolutional neural networks. *ArXiv e-prints*, 2015.

Pachet, F., Papadopoulos, A. and Roy, P. Sampling variations of sequences for structured music generation. In *ISMIR*, 167–173. 2017.

Rosenblatt, F. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Tech. rep., Cornell Aeronautical Lab Inc Buffalo NY, 1961.

Sola, J. and Sevilla, J. Importance of input data normalization for the application of neural networks to complex industrial problems. *IEEE Transactions on nuclear science*, Vol. 44(3), 1464–1468, 1997.

Srivastava. A simple way to prevent neural network from overfitting. *Journal of Machine Learning Research*, Vol. 15, 1929–1958, 2014.

Wassermann, K. C., Eng, K., Verschure, P. F. and Manzolli, J. Live soundscape composition based on synthetic emotions. *IEEE MultiMedia*, Vol. 10(4), 82–90, 2003.

William. Pyposer, a composer based on probabilities, 2018.

Xenakis, I. Analogique. 1958.

Zeiler, M. D. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

Out of the night that covers me,
Black as the pit from pole to pole,
I thank whatever gods may be
For my unconquerable soul.

William Ernest Henley