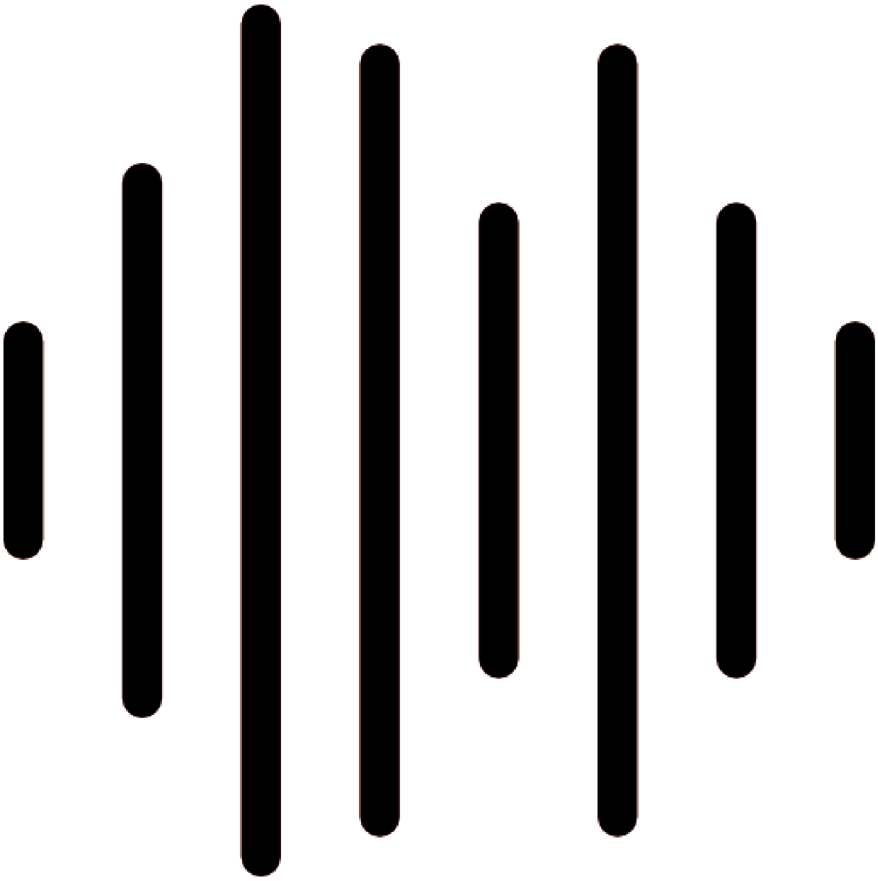
Research Document

Generating music using AI



FSound free icon: Flaticon.com

Date	**22-05-2022**
Version	**0.3**
State	**In progress**
Author	**Mickey Krekels**
Class	**RB04**

VERSION HISTORY

Version	Date	Author(s)	Changes	Status
0.1	01-04-2022	Mickey Krekels	Added the main structure of the document.	In progress
0.2	07-05-2022	Mickey Krekels	Added chapters 3.1 , 3.2 and 3.3	In progress
0.3	25-05-2022	Mickey Krekels	Added chapters 3.4	In progress

TABLE OF CONTENT

ABSTRACT

This research report is made for the “Emerging Trends” assignment provided by Fontys on canvas. In this assignment, the student may choose from certain emerging IT topics, for example, Blockchain, Programming Paradigms, Quantum Computing and Artificial Intelligence and Machine Learning.

For this report, the topic of Artificial Intelligence and Machine Learning is chosen. With the aim of researching if music generation is possible with Artificial Intelligence. Because of this subject, the main research question became “*How is it possible to generate music with machine learning and neural networks?”.*

All of the sub-questions will answer this main question, and the finding will be explained in each chapter and sub conclusion. These chapters are part of a step by step take on how to create music with machine learning and AI. In the end, the main question will be answered in the closing conclusion.

GLOSSARY

ACRONYMS AND ABBREVIATIONS

Abbreviation	Meaning
MoSCoW	Must have, Should have, Could have, Won't have
VM	Virtual Machine
CNN	Convolutional Neural Network
AI	Artificial Intelligence
ML	Machine Learning

INTRODUCTION

I am a third-year student at Fontys university of applied sciences and studying software ICT. During my specialization on Fontys, I studied the subject of AI with the main focus set on neural networks. This project gave me a chance to learn new types of machine learning, this being generating music using AI.

But why is AI or Machine learning so important in this current day? Machine learning is a fast-growing branch within the software and data science sector, it is used in almost all of the top companies that have a big focus on tech, data, and IT.

The document is structured in separate chapters, each containing important parts of research, steps, and problems that occurred during the development of the project. At the end of the report, there will be a conclusion where I will describe the final result.

1 Generating music using artificial intelligence

1.1 Context

Artificial Intelligence (AI) is currently a hot topic in the business industry. It allows for bigger data understanding and manipulation than ever before. Where there is lots of data there is a place for AI and machine learning (ML) to be used. Because of this, it can be used in lots of tasks that previously structured human-written code could not do. The problem that this report will show is that even the generation of music can be done using the newest AI and ML algorithms and techniques.

1.2 Problem description

There are a lot of topics within Artificial Intelligence and Machine Learning that can be researched. From simple classification such as Decision Tree algorithms to more advanced neural network projects.

From my previous specialization semester in Artificial Intelligence, I learned the basics of working with neural networks. For this research, I would like to continue working on this skill. Therefore the main goal is to make a demo project that generates music using a trained neural network.

This technology could provide the music industry with an opportunity with a cheap and fast alternative to writing music.

1.3 Deliverables

The delivery of this research will be a demo project, where the network is able to generate a musical sound file. For documentation purposes, I am going to use Jupyter Notebook (1)*, which is a Python framework for documenting code. This notebook delivery will explain all the training steps, optimization techniques and the end result.*

2 RESEARCH QUESTIONS

In this part of the document, I will describe the most relevant research questions. This will be done by using the Dot Framework research methodology (2).

2.1 MAIN RESEARCH QUESTION

2.1.1 How is it possible to generate music with machine learning and neural networks?

2.2 SUB QUESTIONS

To provide an answer to the main question these sub-questions are required.

2.2.1 How to create a training set for the network?

2.2.2 How to create a validation set for the network?

2.2.3 How is the data structured?

2.2.4 What are the common techniques/algorithms used for this type of network?

2.2.5 How to create a music generation network demo?

2.2.6 How to test the quality of the generated result?

See the table below for the linked category and methods for each of the sub-questions.

Dot Framework research methodology		
Sub Question	**Method**	**Category**
2.2.1	Data analytics	**Lab**
2.2.2	Data analytics	**Lab**
2.2.3	Available product analysis	**Library**
2.2.4	Available product analysis	**Library , Field**
2.2.5	Literature study , Prototyping	**Library , Workshop**
2.2.6	Prototyping	**Workshop**

3 INITIAL PHASE

3.1 THE TRAINING DATA FOR THE NETWORK

Before using a machine learning algorithm, a big amount of training data needs to be acquired. This data will be based on what the output of the network must provide. A popular website for gathering this training data is called “Kaggle” (3), it provides lots of different datasets including music-related training sets.

Such as the Classical music style data set “classical-music-midi” (4) , this collection offers 19 great composers' pieces of classical piano midi files. But the genre it offers is only classic music. This is good if you want to train a classic music generator, but for this research a broader spectrum is preferable.

The Github platform also provides lots of public use datasets, SigSep (5) is also a database set that contains different examples of music with varied instruments. This would be a great option for generating different styles.

But there is also a second option and that is training the network personally. This is as simple as choosing the best version of generated music out of the output and using it as a base for the next generation. The own side is that this can take a lot of time, and human-made errors are always a risk. Therefore this is not ideal for training, the reason that this is mentioned is that sometimes data is not always available, and this is the second-best option in that case.

Conclusion

There are a lot of different options to choose from when it comes to music datasets. The most important aspect when deciding on the right data is what would the preferred output be. In the field of AI, the training set shapes the network to perform the best similar output (This is never 100% accurate but the more you train the network the closer it gets).

Because of these reasons, the dataset SigSep (5) is the most promising. It contains more different music styles and instruments than the other options.

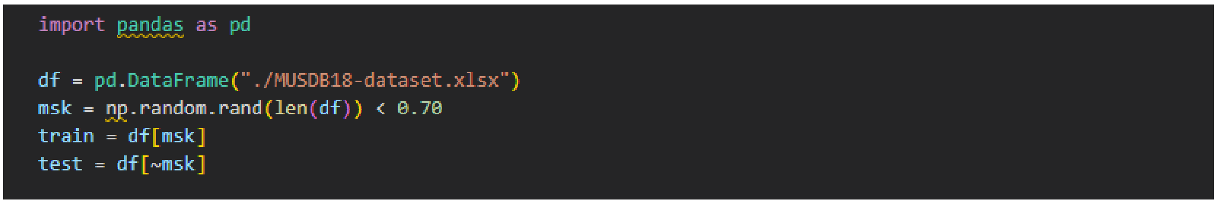
With this decision, the sub-question 2.2.1 “How to create a training set for the network” is answered.

3.2 THE VALIDATION SET FOR THE NETWORK

To create a validation set we need to split the training data. In the folwing article “Splitting a Dataset into Train and Test Sets” (6) the author mentioned that depending on the size of the dataset the spitting ratio can vary. If the n factor (size of the data) is <10.000 the ratio is 70:30 split and around ≈100.000 this can be 80:20. The size of the SigSep dataset is around 150 music tracks and ~10 hours in duration. For this project, we will generate segments of music of around 10 seconds. This would mean that we have ~3600, so the 70:30 split is preferable for this project.

This means using 70% of the original set as training data and the other 30% as validation data, this technique ensures that the algorithm is not tested with the same training data.

In other to spilt this data the Python library “Pandas” (7) , in figure 1 this code is shown as an example.



```
import pandas as pd

df = pd.DataFrame("./MUSDB18-dataset.xlsx")
msk = np.random.rand(len(df)) < 0.70
train = df[msk]
test = df[~msk]
```

Figure 1 splitting the dataset into 80% train and 20% test set

With this step, the validation set is completed and can be used to test the precision of our network.

Conclusion

In order to create a validation set, the training set needs to be split into a 70:30 ratio. The library used for this process in python is called “Pandas” (7). Whit these steps the validation set of this research can be made.

In this chapter, the answer to the sub-question 2.2.2 “How to create a validation set for the network ” is provided and explained.

3.3 THE DATA STRUCTURE

In the machine-learning field is inspecting/cleaning the data structure the most important step. Most all non-cleaned data is not ready to use as input, there are always empty columns and noise that can lower the overall accuracy of the network.

The official term for this approach is called “Exploratory Data Analysis”, the goal is to visualize and inspect the data. With this newly found information, cleaning data and making it input ready makes it much easier.

For this type of AI, these steps are not needed, The only task for this part is to split the music files into smaller fragments for easier training. And remove files the ones where there is no musical value contained. And finally, in the next step, we convert the file into a useable input for the given network. For this, the conversion from sound into a spectrogram format (8) is best suited. (see figure 3 for a list of all these steps)

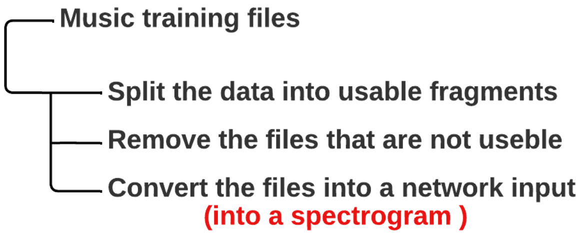

Music training files
Split the data into usable fragments
Remove the files that are not useble
Convert the files into a network input
(into a spectrogram)

Figure 2 cleaning and reformation steps

Conclusion

3.4 MUSIC GENERATION ALGORITHM TYPES

There are a lot of types to consider based on the task you want the AI or Machine learning algorithm to train on.

One of these is the **Genetic Algorithm** which focuses on training by evolution. This means that only the networks with the best results can give their genes to the new generation with the use of the Order Crossover (9) algorithm. In order to provide mutation within these genes, the algoritm Mutation (10) is used. This process provided the network to perform better with each new generation (For a visual overview of these algorithms see figure 2 below).

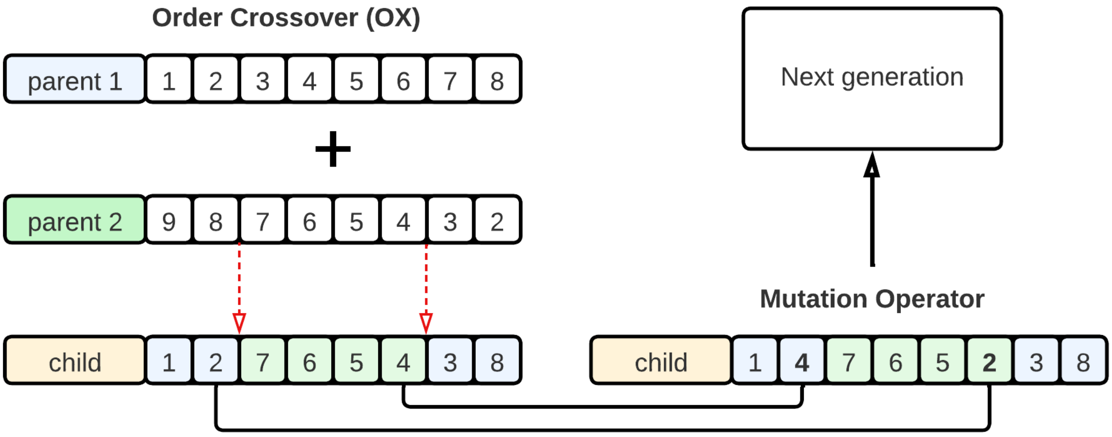


Figure 3 algorithms mutation and ox crossover visualization

The problem with this type is that the validation part is difficult. Each of the musical outputs must be manually graded, which makes it a tedious and long task of training.

The model **WaveNet** (11) is on the other hand specialized for the generation of sound. The network is based on a **Convolutional Neural Network** (CNN), these network layers have different expanding factors, where each of the values is changed for every layer based on the precision of the score when training (For a visual overview of these algorithms see figure 3 below).

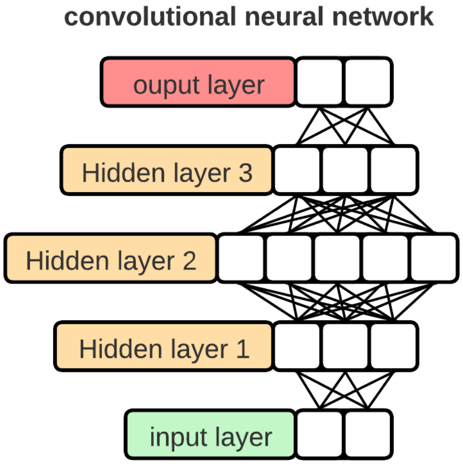


Figure 4 convolutional neural network structure example

Wave net could be a good option for generation music, but the overall implementation is not well documented in Python. Because of this, another CNN model type would be better and easier to implement.

Another popular CNN model is called MuseGAN, this model has a lot of public repositories available on Github. This version is compatible with the library PyTorch (12) which makes it easy to change the structure of the network (adding extra layers etc.). Therefore this library is truly perfect for this research.

Conclusion

We looked at 3 possible ways to train an AI/machine learning algorithm to generate music. The Genetic Algorithm was a good way to train accurate on generating melodies, the downside was the training part must be done manually. The WaveNet model was great for accuracy but de documentation was little to non-existing. Which makes it difficult to implement with a Python-based project. And lastly, the model MuseGAN has also high precision and is well documented. Therefore the MuseGAN model is the best-suited network for this project. For a more visual overview see the table below.

Model Type	Pros	Cons
Genetic Algorithm	Easy to use	Manual model training
WaveNet (CNN)	High accuracy and precision	Documentation is laking
MuseGAN (CNN)	High accuracy and precision, good documented, can be used with PyTorch	None

In this chapter, the answer to the sub-question 2.2.4 “What are the common techniques/algorithms used for this type of network?” is provided and explained.

3.5 CREATING A MUSIC GENERATOR

Conclusion

3.6 TESTING THE QUALITY OF THE MUSIC GENERATOR

Conclusion

4 CONCLUSION

(this chapter will be added at a later date)

5 BIBLIOGRAPHY

1. **Ingargiola, Antonino.** What is the Jupyter Notebook? *jupyter-notebook-beginner-guide.* [Online] 2015. https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html.

2. **The DOT Framework. *ictresearchmethods.* [Online] https://ictresearchmethods.nl/The_DOT_Framework.**

3. Start with more than a blinking cursor. *kaggle.* [Online] https://www.kaggle.com/.

4. Rakshit, Soumik. Classical Music MIDI. *kaggle.* [Online] https://www.kaggle.com/datasets/soumikrakshit/classical-music-midi.

5. Rafii, Zafar, et al. sigsep MUSDB18. *sigsep.github.* [Online] https://sigsep.github.io/datasets/musdb.html#musdb18-compressed-stems.

6. Tokuç, A. Aylin. Splitting a Dataset into Train and Test Sets. *baeldung.* [Online] 14 1 2021. https://www.baeldung.com/cs/train-test-datasets-ratio.

7. Getting started. *pandas.* [Online] https://pandas.pydata.org/.

8. Spectrogram. *wikipedia.* [Online] https://nl.wikipedia.org/wiki/Spectrogram.

9. Holland, John , Eshelman, Larry J. en Gwiazda, Tomasz D. Crossover (genetic algorithm). *wikipedia.* [Online] https://en.wikipedia.org/wiki/Crossover_(genetic_algorithm).

10. Holland, John. Mutation (genetic algorithm). *wikipedia.* [Online] https://en.wikipedia.org/wiki/Mutation_(genetic_algorithm).

11. Oord, Aäron van den en Dieleman, Sander . WaveNet: A generative model for raw audio. *deepmind.* [Online] 8 9 2016. https://www.deepmind.com/blog/wavenet-a-generative-model-for-raw-audio.

12. From Research To Production. *pytorch.* [Online] https://pytorch.org.