

# 16153014\_FYP\_Project

-----

This project investigates the application of neural networks to music sequence composition, contained in this project are two neural network models that attempt to achieve this goal. The first, `autoencoder.py` is a simple autoencoder model that will generate new sequences, the second, `variational_ae.py` is an improvement on the autoencoder model to try and achieve better results.

For the midi datasets created for this project the majority of midi files came from this link

[https://mega.nz/#!Elg1TA7T!MXEZPzq9s9YObiUcMCoNQJmCbawZqzAkHzY4Ym6Gs\\_Q](https://mega.nz/#!Elg1TA7T!MXEZPzq9s9YObiUcMCoNQJmCbawZqzAkHzY4Ym6Gs_Q)

Below is a description of the system scripts and their usage.

## -----SYSTEM REQUIREMENTS-----

windows: Note: If using on a linux system file paths must be changed

Python3.6

Tensorflow\_cpu 2.0

Music21

-----

**process\_midi\_dataset.py:** Description: Script to process supplied midi dataset so it can be used in the neural network models.

Outputs: `-.npz` file containing sequences ready  
for neural network training

**usage:** `process_midi_dataset.py [-h] --dataset DATASET --npz_file NPZ_FILE`  
`--network {AE,VAE} --length LENGTH --trim`  
`{True,False}`

### Process MIDI dataset.

optional arguments:

`-h, --help` show this help message and exit

`--dataset DATASET` Name of dataset in the dataset folder: Usage: `bach_50`

`--npz_file NPZ_FILE` `.npz` MIDI sequence save file: Usage: `<dataset_name.npz>`

`--network {AE,VAE}` The type of network the dataset will be used for: usage: `<AE/VAE>`

--length LENGTH    Length of processed sequences: Usage: <64>  
--trim {True,False} Argument to indicate if the script should process the  
entire songs or only the first sequence of <length>  
notes: Usage: <True/False>

example usage:

```
python process_midi_dataset.py --dataset bach_50 --npz_file bach_s50_sq64.npz  
--network VAE --length 64 --trim False
```

---

---

**process\_similar\_sequences.py:** Description: Script takes in a midi dataset path and a  
single midi file, then searched the dataset  
for sequences similar to the provided midi.

**Outputs:** -.npz file containing all similar  
similar sequences as well as the  
sequences of the single song

usage: process\_similar\_files.py [-h] --dataset DATASET --song\_dir SONG\_DIR  
--npz\_file NPZ\_FILE --length LENGTH

**Process similar sequences of a song from a datasets.**

optional arguments:

-h, --help    show this help message and exit  
--dataset DATASET Path to MIDI dataset folder: Usage:  
                  datasets\<dataset\_name>\  
--song\_dir SONG\_DIR Path to MIDI file to try and match: Usage:  
                  datasets\<dataset\_name>\<song\_name.mid>  
--npz\_file NPZ\_FILE .npz MIDI sequence save file: Usage: <name.npz>  
--length LENGTH    Length of processed sequences: Usage: <64>

example usage:

```
python process_similar_sequences.py --dataset datasets\bach_50\ --song_dir  
datasets\bach_50\air.mid --npz_file similar_sequences.npz --length 64
```

---

---

**autoencoder.py:** Description: Script that contains the base autoencoder model,  
trains the neural network and produces a single generated  
midi sequence

usage: autoencoder.py [-h] --dataset DATASET --latent\_dim LATENT\_DIM --epochs  
EPOCHS

## Autoencoder training script

optional arguments:

- h, --help show this help message and exit
- dataset DATASET Name of the training npz file: Usage: <bach\_s50\_sq64>
- latent\_dim LATENT\_DIM Size of the network latent dimension: Note:  
Recommended to be 1/4 of training sequence size:  
Usage: <12>
- epochs EPOCHS Number of training epochs for the network: Usage: <50>

example usage:

```
python autoencoder.py --dataset bach_s50_sq64 --latent_dim 12 --epochs 100
```

---

---

**variational\_ae.py:** Description: Script that contains the variational autoencoder model, trains the neural network and saves the trained modes and produces three generated midi sequences.

usage: variational\_ae.py [-h] --dataset DATASET --batch\_size BATCH\_SIZE --epochs EPOCHS

## Variational Autoencoder Training Script

optional arguments:

- h, --help show this help message and exit
- dataset DATASET Name of the training npz file: Usage: <bach\_s50\_sq64>
- batch\_size BATCH\_SIZE  
Number of samples to be processed per iteration: Note:  
Recommended to be some multiple of sequence size but  
less than total number of samples: Usage: <12>
- epochs EPOCHS Number of training epochs for the network: Usage: <50>

example usage:

```
python variational_ae.py --dataset bach_s50_sq64 --batch_size 64 --epochs 100
```

---

---

**load\_model.py:** Description: This script is used to load a pre-trained model using a json file and corresponding h5 file.

Autoencoder training script

optional arguments:

-h, --help            show this help message and exit  
--dataset DATASET    Name of the training npz file used during model  
                      training: Usage: <npz file name>  
--model MODEL        Name of model file from models folder: Usage:  
                      <model\_file\_name>  
--weights WEIGHTS     Name of weights file from weights folder : Usage:  
                      <weights\_file\_name>  
--num\_songs NUM\_SONGS  
                      Number of songs to be generated: Usage: <5>

Example usage:

python **load\_model.py** --dataset bach\_s50\_sq64 --model  
generator\_bach\_s50\_sq64\_vae\_model\_26\_4\_2020\_18397 --weights  
generator\_bach\_s50\_sq64\_vae\_weights\_26\_4\_2020\_18397 --num\_songs 10

-----  
-----