Journal of Theoretical and Applied Mechanics, Sofia, Vol.54 (2024) pp. 323-338
INTERDISCIPLINARY TOPICS

# GENERATING NON-MONOPHONIC MUSIC WITH VARIATIONAL AUTOENCODER: A CASE STUDY

IVAN K. YANAKIEV*

*Institue of Mechanics, Bulgarian Academy of Sciences, 1113 Sofia, Bulgaria*

[Received: 20 November 2023. Accepted: 1 October 2024]

**doi: https://doi.org/10.55787/jtams.24.54.3.323**

ABSTRACT: The paper proposes an approach for generating musical pieces based on the Variational Auto Encoder (VAE) and realized in the MATLAB framework. In the work the architecture of the used VAE is described and also its training with dataset from the MAESTRO MIDI Dataset. The aim of the work is to study to what extent the used VAE is suitable for learning and representing musical structure. The resulting pieces show some features of musical structure on a mid-term temporal scale. It is concluded that the model successfully represents traits of general musical structure on the broader timescale as well as some short-scale musical figures. However, it fails to account for time-related musical structure of the long-term musical architectonics.

KEY WORDS: VAE, variational autoencoder, MATLAB, music generation, music entropy, music data processing, midi, AI for music, music neural network.

## 1 INTRODUCTION

VAEs can be categorized as probabilistic generative models with two neural networks (NN) [1,2]. The NN elements are usually named encoder and decoder. The encoder's primary function is to map the input data into a latent space that characterizes the parameters of a variational distribution. Conversely, the decoder operates in an inverse fashion, mapping from the latent space back to the input space, thereby generating or reconstructing data points. Both the encoder and decoder networks are trained simultaneously, leveraging a technique known as "the reparameterization trick". The variance of the noise model can also be learned independently. While VAEs were originally conceived for unsupervised learning, their utility has been demonstrated in diverse contexts, including semi-supervised and supervised learning scenarios. Another major direction of their use is treating them as generative models by dropping the encoder and feeding the decoder a random data with the formal characteristics

*Corresponding author e-mail: yanakiev.ivan@imbm.bas.bg

of the latent space. This transformation of the VAE NN into a generative NN is the subject of the research presented in this paper [2].

### 1.1 Image Generation and MNIST Database

One anthological example of using VAE is the generation of handwritten digit based on the MNIST (Modified National Institute of Standards and Technology) dataset [1] which give the opportunity to experiment with different NN because of the dataset diversity and relatively small image size (28×28 pixels). It contains 60000 training and 10000 testing images.

The symbiosis of NN and the MNIST dataset is used for displaying capabilities of the NN architecture to generate new images. The model discussed in this text is a VAE designed and trained to encode into its latent space and generating new images by decoding a sample from a random distribution as if it was taken from the latent space of the VAE. The discussed example of a generative model of interest to the current research is the one described by MATHWORKS TEAM [2].

The result has some properties that might be of special interest for generating musical examples:

— the VAE is able to synthesize almost all of the distinct features of the numbers very well;

— the VAE generates a very smooth transitional output of the generated images' features that are proprietary to one number abut are assigned to another number (due to the randomness of the distribution, with which the latent space has been substituted)

These features allow us to consider the option this VAE to be harnessed into the generative process in art forms – for example visual and musical.

### 1.2 The Latent space and smooth transition

The most prominent example is using VAE in the image domain is its implementations for generating new human faces [3, 4]. The results of the VAE for facial generations are images with features that are virtually recognizable as real human face, although the faces are all generated by the NN. Based on the general ability of VAE to capture specific traits there are certain VAE models that are designed to transfer the specific traits. This is possible because of the way the latent space works – it clusters together close features, detected by the convolutional layers of the encoder network.

This ability is also used in the musical domain [5]. The idea to map certain structural features in the temporal or pitch-class domains into the latent space and then

randomly sample from the latent distribution is a powerful method for generating new musical examples.

## 2 Musical Application of the VAE Concept

### 2.1 MAESTRO dataset

In the proposed model the musical data was selected from the MAESTRO (MIDI and Audio Edited for Synchronous Tracks and Organization) [6, 7] dataset. This dataset contains 200 hours of music written and performed for piano exclusively. It consists of two parts – MIDI files and WAVE files. For the current model only the MIDI part was of any practical interest[1].

### 2.2 Application

The MAESTRO Dataset was used in numerous NN for music processing and generation. It is valued because of its feature – it represents human performance traits as it is in fact MIDI recording of live human performance. Taking into account that these recordings were made at a competition – we may count at least to a reasonable extend – the exemplary pieces contained in the dataset are of high artistic quality [6].

Some of the most recent integrations of the MAESTRO Dataset into research are listed in Table 1. These include projects researching music transcription, music generation (symbolic and WAVE), generations of music arrangements, a BERT based musical model etc. It can be concluded that the MAESTRO Dataset is well recognized and suitable up-to-date option on top of which base research can be grounded. This allows also a viable way for making a comparison and staying up-to-date with the current state of the research with the music generation with NN field. The data presented in Table 1 is retrieved form the Papers with code website [8] which list 78 projects that use MAESTRO dataset spanning a five-year period (2018-2023).

### 2.3 Data preprocessing

The data of the Maestro Dataset is structured in such a way that it is suitable to be imported in MATLAB. However, MATLAB does not have its own midi processing functions. In order to solve this limitation a very useful library was implemented – MIDI Toolbox 2.0 [9, 10]. This library enables us to treat the MIDI files as matrices (nmat or note matrix) with size [7 n] and each column extracts relevant information for each MIDI event from the MIDI table. The columns of interest are the MIDI time onset (in beats or in ms), the MIDI pitch number, the duration of the note and the

[1] The audio wave parts possess the potential to alleviate the transition from discrete MIDI data towards continuous wave signal processing.

Table 1: Recent use of Maestro dataset [8]

| Paper | Date |
|---|---|
| MusicBERT: Symbolic Music Understanding with Large-Scale Pre-Training | 10 Jun 2021 |
| A Lightweight Instrument-Agnostic Model for Polyphonic Note Transcription and Multipitch Estimation | 18 Mar 2022 |
| High-resolution Piano Transcription with Pedals by Regressing Onset and Offset Times | 5 Oct 2020 |
| MT3: Multi-Task Multitrack Music Transcription | 4 Nov 2021 |
| Musika! Fast Infinite Waveform Music Generation | 18 Aug 2022 |
| Byte Pair Encoding for Symbolic Music | 27 Jan 2023 |
| MusPy: A Toolkit for Symbolic Music Generation | 5 Aug 2020 |
| LakhNES: Improving multi-instrumental music generation with cross-domain pre-training | 10 Jul 2019 |
| POP909: A Pop-song Dataset for Music Arrangement Generation | 17 Aug 2020 |
| MelNet: A Generative Model for Audio in the Frequency Domain | 4 Jun 2019 |

velocity (loudness) of the midi pitch. From the full nmat-s only columns 1, 3, 6 and 7 are extracted for preparing the input data.

For the training of the VAE from the whole dataset of 60 composers only several composers with their corresponding examples were selected. They were chosen on musicological basis – the most similar in style to one another and closest to the musical characteristics of the Galant style were chosen. Table 2 gives a list of the composers and the examples that are included in the research data.

From the whole 1276 examples only 117 were used. This limited number of data required data augmentation to take place. It was done by transposing each piece chromatically into all 12 cardinal tonalities chromatically. This resulted into a training set of 1404 pieces. Further steps for augmenting the data were scaling its musical time (speed of the whole example), duration (the length of each note) and its onset (the space or the duration of the silence between each note). With each dimension (time, onset and scale) varied in maximum 8 versions – (linespace vector form 0.25 with step of 0.25 up to 2) gave a total of 288 variations of each example. Thus from 117 examples we end with 26 676 input examples (values comparable to the proposed by MATHWORKS size of input data for the VAE).

Table 2: Composers and examples

| No in the dataset | Composer | Number of examples |
|---|---|---|
| 8 | Domenico Scarlatti | 31 |
| 30 | Johann Christian Fischer / Wolfgang Amadeus Mozart | 1 |
| 39 | Joseph Haydn | 40 |
| 45 | Muzio Clementi | 6 |
| 48 | Orlando Gibbons | 1 |
| 60 | Wolfgang Amadeus Mozart | 38 |
| | 6 composers | 117 examples |

The data was then rescaled so that each column independently took values from 0 to 1. The maximum and the minimum values were saved for reconstruction purposes. Finally, the resultant matrices were zero padded when needed to fit the maximum height of the nearest power of 2 to the length of the longest piece (8192).

Some of the test runs of the NN used 2-dimensional cell structure with size of $117 \times 12$ cells. The final augmentation of the data was externalized. Each cell contained matrices with size [4 8192] scaled from 0 to 1 zero that represent the onset time in ms, the MIDI pitch value, the velocity and the duration of the pieces.

### 2.4 Structure of the VAE

The designed VAE is an adaptation of an image generation VAE trained with the MNIST dataset [2]. As a substitute to the $28 \times 28 \times 1$ black and white images of the hand written digits the data from the MIDI files of the MAESTRO Dataset extracted by the readmidi function of the MIDI Toolbox library is used. The input data is extracted, augmented (described in the previous section) rescaled and padded to size [4 8192] from the original nmat representation [7 length_of_example]. The prepared data is fed to the VAE as batch tensors with size [4 8192 Batch_size, Piece_index]. MatLab's capability of using the datastore function were implemented so that the input data can exceeds the maximum of the allocated memory. This enabled to do the tests with more pieces from the database.

The encoder and the decoder parts have six convolutional layer – relu dropout layer complexes, that shrinks the length dimension of the data 2 times with every convolutional layer, resulting in a 64 times reduction of the length of the input data.

### 2.5 Training

The network is trained iteratively by a loop defined by the number of epochs. The data is fed to the training chain in minibatches, which size is set to maximize the capabilities of the computational hardware. A gradient descend method for loss minimization with adam optimizer was implemented.

### 2.6 Learning rate

The prepared VAE use Evidence lower bound (ELBO) loss [11, 12]. It is calculated by summing the reconstruction loss with the Kullback–Leibler (KL) divergence [5] (reparameterization trick [13]).

In the final version of the proposed NN the learning rate is updated by using a cyclical learning rate algorithm which is modified to extend the length of the cycle further as the number of epochs iterate. This resulted in a much faster reach of a minimum of the gradient descend. The implemented solution here is following the suggested algorithm by Sanket Doshi [14].

## 3 Experiments[2]

The NN was trained several times in order to be optimized. For the needs of this text only 5 pivotal test runs will be described. They have in common the following parameters: Number of epochs – 80; Latent space dimensions [128 1 1024 ], Number of latent channels – 20, MiniBatch size – 12[3].

All the results from Test 2–Test 5 were generally tonally stable. The results from Test 1 are without tonal center.

### 3.1 Test 1

MAESTRO Database – All data from Haydn with Train index: augmented by only transposing it from the 12 pitches (40 pieces with 12 variation per piece = 480 data entries.), 80 epochs (Fig. 1).

The error is very low (Fig. 2): between -0.05 and 0.15 and centered around 0.05.

The result sounds random and atonal. The closest resemblance is to a heterophonic piece or a free atonal improvisation without considering any tonal or temporal structure. The data is very close to mapping noise to pitch classes and note lengths.

[2] Results can be found online at: https://github.com/Yanakiev432/VAE-for-generating-musical-examples#vae-for-generating-musical-examples

[3] Some of these parameters were chosen due to the hardware limitations (Batch size, Number of latent channels), others were a result of the manual parameter optimization.

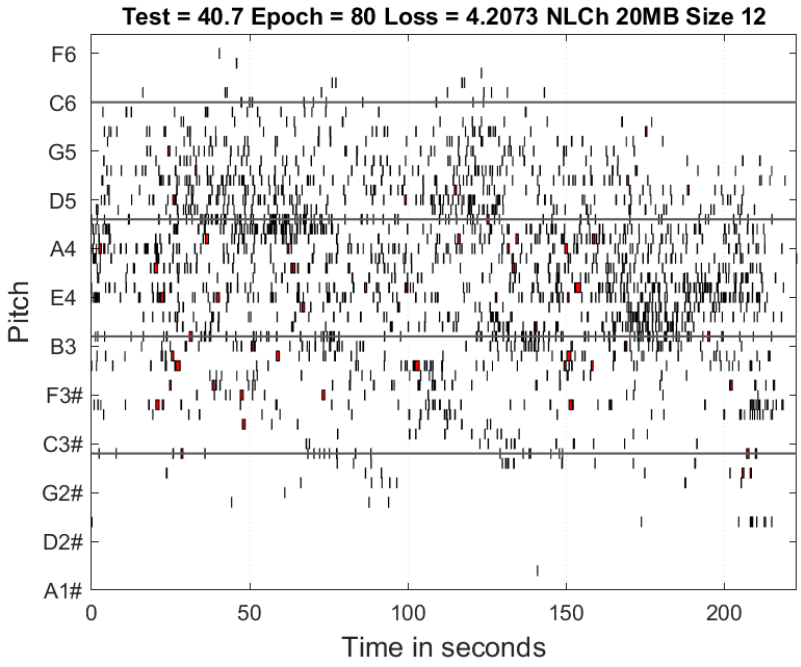


Fig. 1: Pianoroll representation of the generated music piece in Test 1.

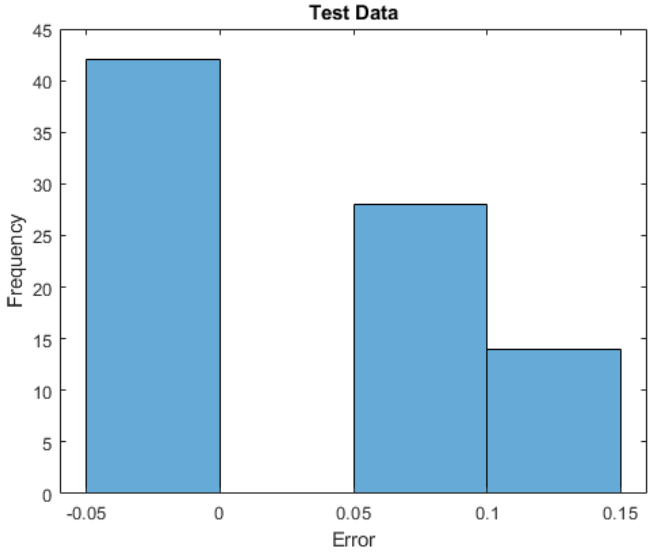


Fig. 2: Histogram of the error of the trained network in Test 1.

### 3.2 TEST 2

MAESTRO Database – all data from Haydn with Training index: augmented by transposing it and changing the length, duration, pitch in 4 different variations (40 pieces with 12 variation per piece varied in 4 different values IN 3 different parameters = 5760 data entries.), 80 epochs.

The resultant pieces are very similar to one another and have lengths of around 4:30 min.

The visually presented results in Fig. 3 are comparably more structured regarding Test 1 results in Fig. 1. The loss of the network is 3.3147, the error of the network is shown in Fig. 4.

We can see here that the error is very small – min 0.03, max 0.055 with the distribution skewed to the left side (most examples have low error). The error distribution is not zero-centered, but positively shifted.

The resultant examples sound more diatonically with certain degree of randomness. Some generalized structures can be herd, like modulation to the double dom-

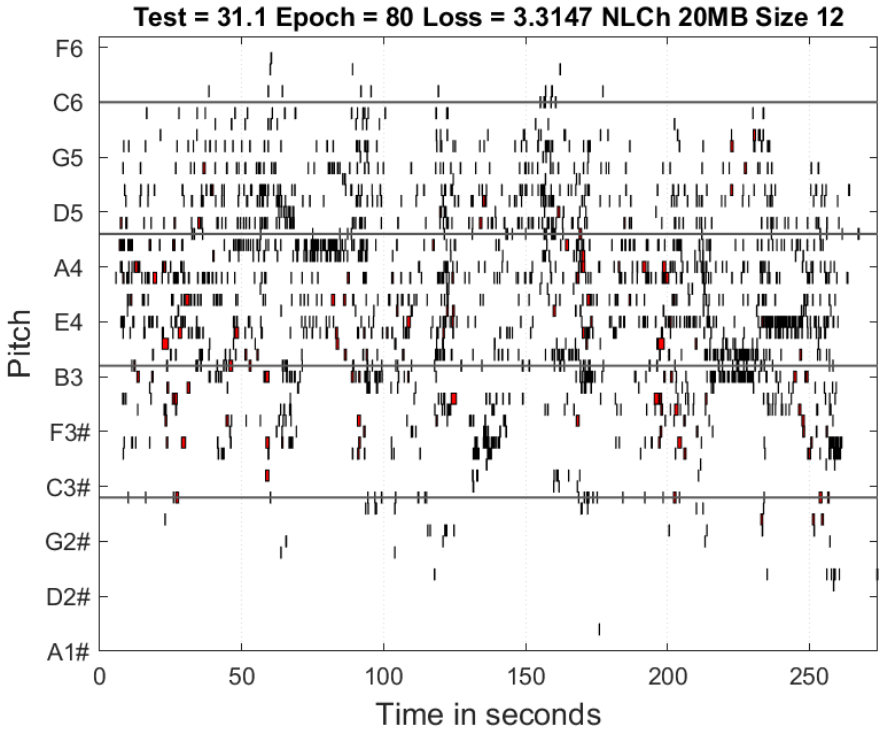


Fig. 3: Pianoroll representation of the generated music piece in Test 2.

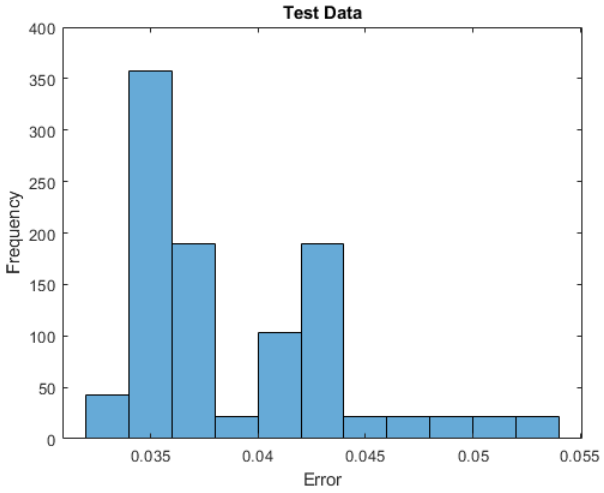


Fig. 4: Histogram of the error of the trained network in Test 2.

inant, some movements in the base iii-iv-v-vi; there can be heard a tendency to distinguish some general fest-locker contrast and the locker sections are characterized with more chromatic randomness as the locker sections are more diatonic. In the fest structures the randomness of the pitch is replaced by multiple repetitions of one tone (in diatonic relations to the other ones).

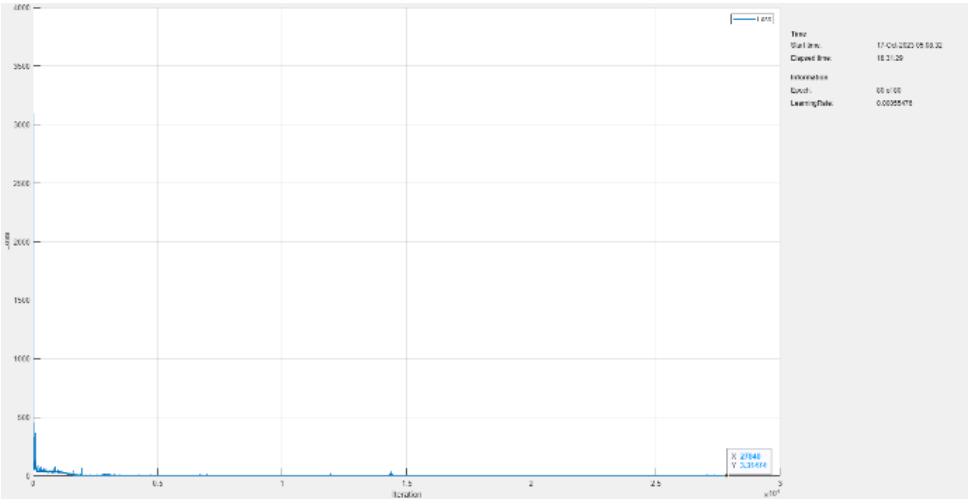

Fig. 5: Learning rate of the network in Test 2.

Generally speaking, some structure is learned by the network after augmenting the data (Fig. 5). However, all the examples are sounding very similar and have similar lengths. This may be a side effect of some overfitting of the model.

### 3.3 TEST 3

MAESTRO Database – all data from Haydn. Augmented by transposing it and changing the length, duration, pitch in 4 different variations (40 pieces with 12 variation per piece varied in 8 different values in 3 different parameters = 11520 data entries.) – doubling the input data, 80 epochs (Fig. 6).

The pieces are similar to the ones of Test 2, have length of about 3:33 min.

The results are very similar to those of Test 2. The main difference is that as the input data was diversified to include shorter divisions of the note values, dynamic range and tempo, the resultant pieces were more metrically diverse, and to some extend also more expressively diverse. This allowed some of the learned by the network generalized harmonical structures to be expressed in a clearer way.

As the general limits of the error of the network remained the same with the results of Test 2 (min 0.03, max 0.055) the only difference was that there is more data to fit

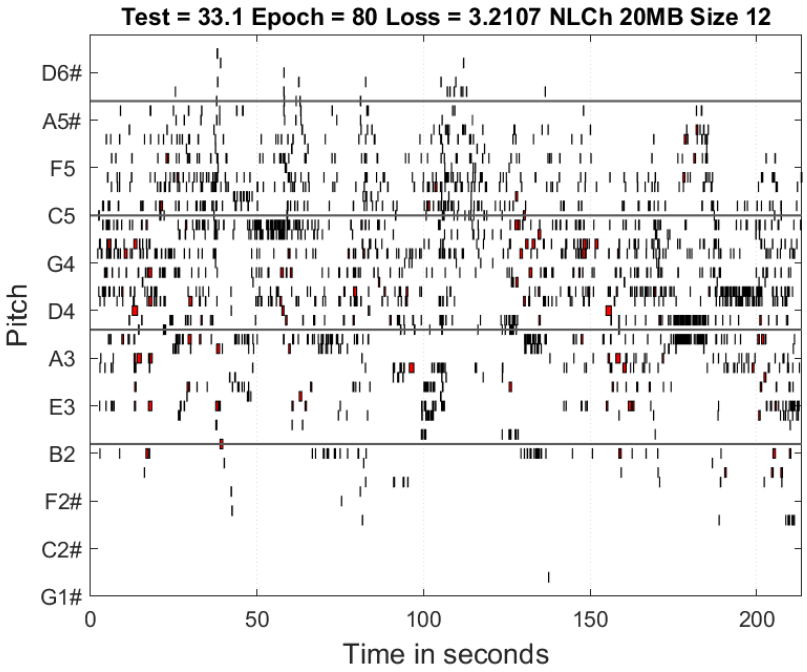


Fig. 6: Pianoroll representation of the generated music piece in Test 3.

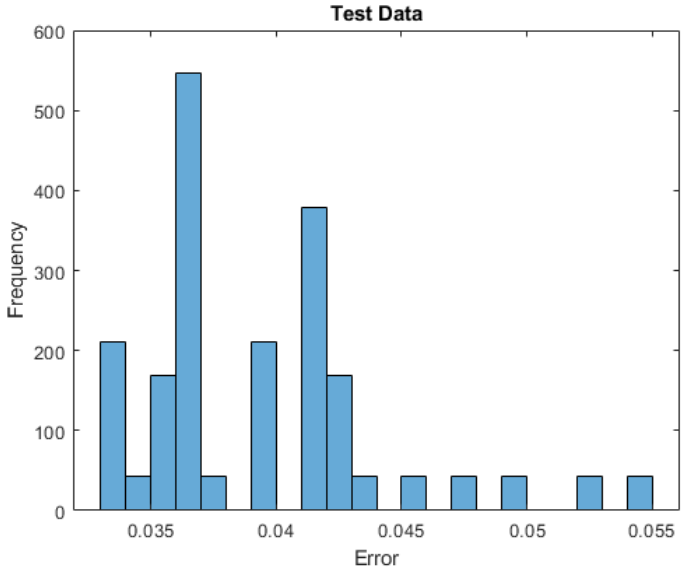


Fig. 7: Histogram of the error of the trained network in Test 3.

between the limits of the distribution (Fig. 7).

Some interesting differences is that now with the augmented time domain of the input data some of the repeated tones started to sound as ostinato octave complexes or trill plus octave complex – a distinct trait of the keyboard music of the Galant and Classical style.

### 3.4 Test 4

MAESTRO Database – all data from Haydn, Scarlatti, Mozart, Gibbons and Clementi (see Table 2) with Train index: augmented by transposing it and changing the length, duration, pitch in 4 different variations (117 pieces with 12 variation per piece varied in 8 different values in 3 different parameters = 33696 data entries.), 8 epochs (Fig. 8).

The resultant examples are more diatonic than random, have the same general length of about 11:34 minutes. They sound very diatonic and tonal, as if fixed to the key of C major. The error is distributed towards the extremes with two peaks in the negative and the positive ends of the distribution (Fig. 9).

It is very interesting that some very specific for the Galant Style cadential moments are very well represented by the results cadential moments are very well represented by the network. The results sound as if they are blurred by some noise translated to midi pitch classes.

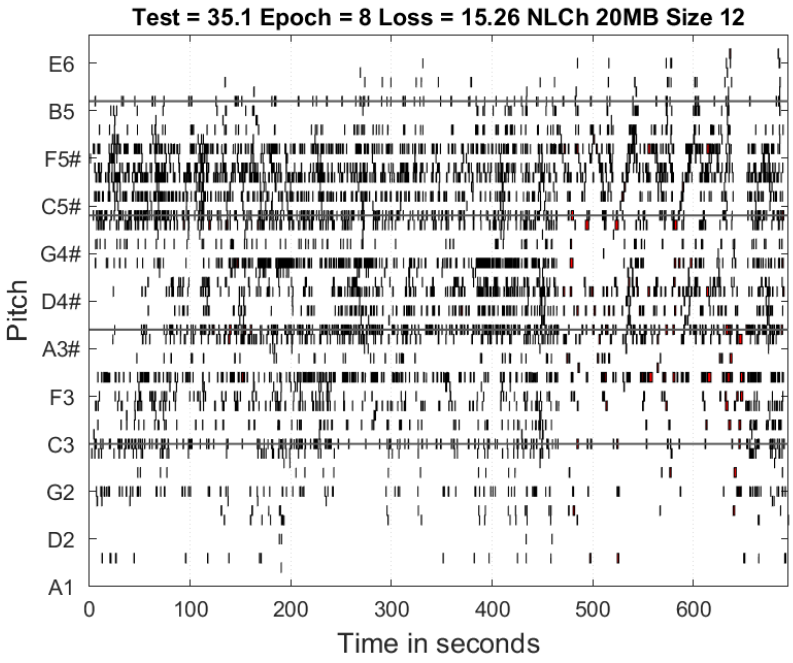


Fig. 8: Pianoroll representation of the generated music piece in Test 4.

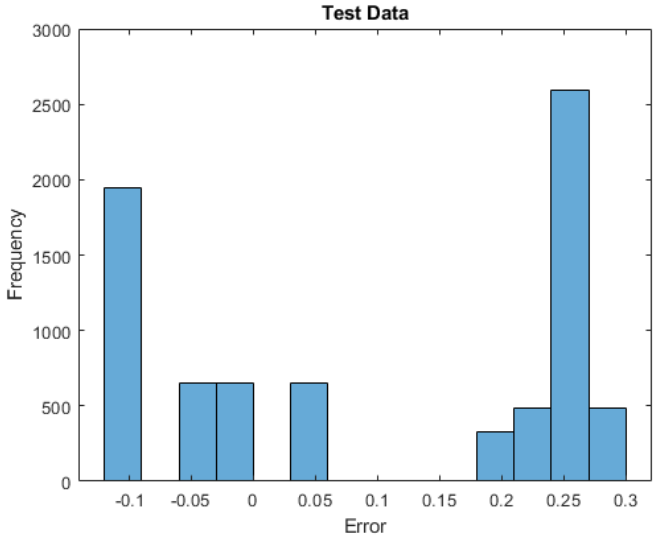


Fig. 9: Histogram of the error of the trained network in Test 4.

The importance of this test that it showed that compared with a more extensive training with the same parameters (see Fig. 11) the 8 epochs sufficed to generate the similar error to the 10 times longer training.

### 3.5 TEST 5

MAESTRO Dataset – all data from Haydn, Scarlatti, Mozart, Gibbons and Clementi (see Table 2) with Train index: augmented by transposing it and changing the length, duration, pitch in 4 different variations (117 pieces with 12 variation per piece varied in 8 different values in 3 different parameters = 33696 data entries.), 80 epochs (Fig. 10).

The length of the generated pieces is around 11:30 min. The error of the last two trained networks (Test 4 and Test 5) is almost identical (Fig. 11), the results are somewhat different, nevertheless. Generally speaking – the loss is too high – around 25 for Test 4 and around 15 for Test 5, so that distinct structures can be heard. The preliminary conclusion is that there are too few data in order the model to build a good representation of the dataset. However, the computation for the Test 5 was to expensive – it took 110 hours to finish the training and the resultant error is identical

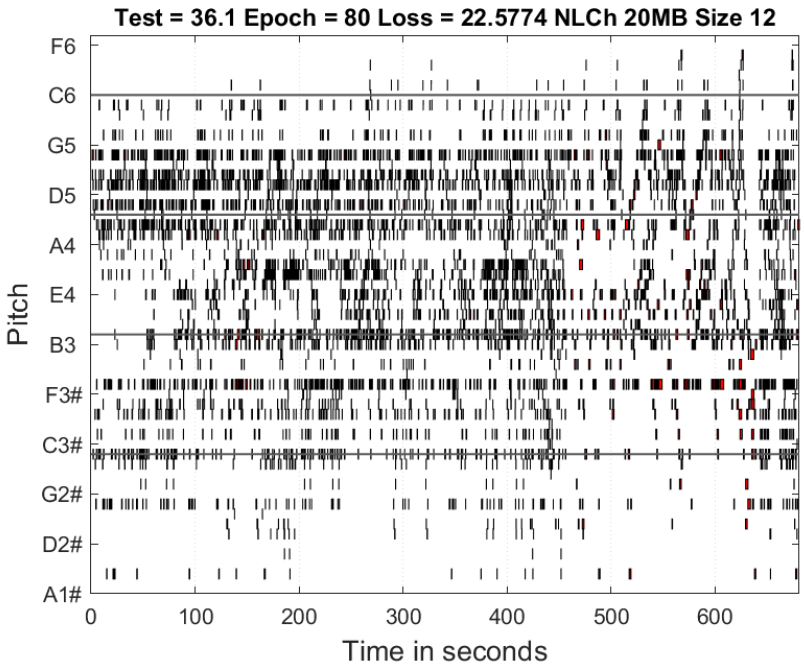


Fig. 10: Pianoroll representation of the generated music piece in Test 5.

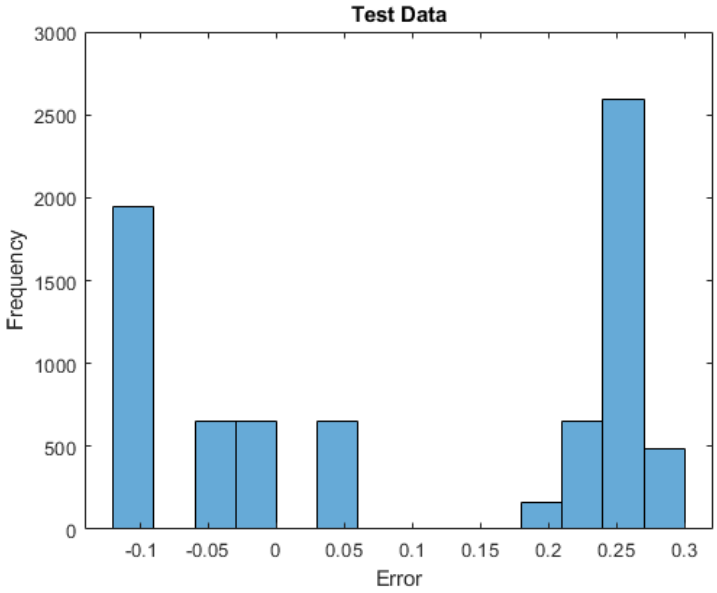


Fig. 11: Histogram of the error of the trained network in Test 3.

with the 10 hour 8 epoch training session of Test 4. There is no big difference in the musical characteristics of the generated pieces either.

One distinct feature of the generated examples is the heterophonic musical structure of the piece – they sound as if at least two musical thoughts are happening in parallel. Vertically there is a distinction between the higher voice (voices) and the bass. This may be result of the higher loss value of the network. As it was expressed by the reviewed data from the conducted tests the higher loss is generally due to not enough similar data that hinders the generalization process of the encoder network.

## 4 Conclusions

The VAE provides a viable option for generating musical examples after training on preexisting input data. The network is able to represent some musical structures, but lacks the longer temporal structural relation of musical gestures. The network was able to represent in its own way several distinct traits of the Galant and Classical style: cadences, musical formulae, however the rather noisy output is one of its greatest disadvantages. The common trait of all the tests – the persistence of existing noise can be associated with the VAE general high rate of blurriness of the generated images, reported in the image generation models.

Another very distinct trait is that with the augmentation of the data and the ex-

istence of more variations of an original data input the model was able to get better general characteristics of the musical style (Galant and early Classical) like melodic formulae in the higher voice as well as bass structures.

## 5 Further Research

By getting more convolutional layers some of the more distinct features in the lo-resolution time domain may be captured in a better way. The network's parameters can be further modified to have a bigger latent space (more features) or bigger batch sizes in order to better generalize the whole data set.

Of course, the data augmentation must be further elaborated by including more varied examples of the original data and also – further diversifying the data by including partial representations of the pieces. This idea of including partial representation, combined with some feature extraction algorithm based on harmonic structure identification mechanism may become a valuable modification for further instances of the model.

## Acknowledgment

The author expresses his gratitude to the National Science Fund and the Bulgarian Academy of Sciences program "Young Scientists and Post-Doctorals – 2" for the provided funding.

## References

[1] L. Deng (2012) The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. *IEEE Signal Processing Magazine* **29**(6) 141-142.

[2] MathWorks (accessed Nov. 2023) Train Variational Autoencoder (VAE) to Generate Images. https://www.mathworks.com/help/deeplearning/ug/train-a-variational-autoencoder-vae-to-generate-images.html.

[3] X. Hou, K. Sun, L. Shen, G. Qiu (2019) Improving variational autoencoder with deep feature consistent and generative adversarial training. *Neurocomputing* **341** 183-194.

[4] Generate Anime Character with Variational Auto-encoder (accessed Nov. 2023) https://medium.com/@wuga/generate-anime-character-with-variational-auto-encoder-81e3134d1439.

[5] H.H. Tan (accessed Nov. 2023) VAE In Symbolic Music Modelling. https://gudgud96.github.io/2020/01/26/vae-symbolic-music/.

[6] The MAESTRO Dataset (accessed Nov. 2023) https://magenta.tensorflow.org/datasets/maestro.

[7] C. HAWTHORNE, A. STASYUK, A. ROBERTS, I. SIMON, C.-ZHI A. HUANG, S. DIELEMAN, E. ELSEN, J. ENGEL, D. ECK (2018) Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset. https://doi.org/10.48550/arXiv.1810.12247.

[8] MAESTRO https://paperswithcode.com/dataset/maestro.

[9] MIDI TOOLBOX (accessed Nov. 2023) https://www.jyu.fi/hytk/fi/laitokset/mutku/en/research/mterals/miditoolbox.

[10] T. EEROLA, P. TOIVIAINEN (2004). MIDI Toolbox: MATLAB Tools for Music Research. University of Jyväskylä: Kopijyvä, Jyväskylä, Finland, http://www.jyu.fi/msica/miditoolbox/.

[11] Y. JIANG (accessed Nov. 2023) ELBO What & Why. https://yunfanj.com/blog/2021/01/11/ELBO.html.

[12] M.N. BERNSTEIN (accessed Nov. 2023) The evidence lower bound (ELBO), https://mbernste.github.io/posts/elbo/.

[13] G. GUNDERSEN (accessed Nov. 2023) The Reparameterization Trick. https://gregorygundersen.com/blog/2018/04/29/reparameterization/.

[14] S. DOSHI (2021) Cyclical Learning Rates. https://medium.com/analytics-vidhya/cyclical-learning-rates-a922a60e8c04.