Proposal of method for generating musical compositions of different genres

N A Nikitin¹, Y A Orlova¹, V L Rozaliev¹, A S Kuznetsova¹ and V V Gilka¹

¹Volgograd State Technical University, Lenina Ave., 28, Volgograd, 400005, Russia
E-mail: nikitin.nikita@gmail.com

Abstract. The work is dedicated to the research and development of methods and algorithms that automate and support the process of technical creativity, from the point of automated creation of musical compositions with different genres. The method is based on the use of neural networks to predict composition, and also suggests using various models that have been trained by songs in certain genres to improve the quality of the resulting musical composition. This work describes the algorithm itself, the process of collecting and classifying data for training, the process of training a neural network. In addition, it describes the process of choosing a neural network architecture. Also, the work proposes a system architecture for the automated generation of compositions using the proposed algorithm.

1. Introduction

Nowadays, more and more papers are being published aimed at automating the process of creating musical compositions, however, this process is creative, depends on many factors, starting from the experience and mood of the composer, ending with the area of residence and other external factors, so music cannot be created in automatically mode, therefore, the role of the user-composer is very high and we can only talk about the automation of this process. The emotionality that is conveyed by music and paintings is difficult to recognize [1]. Although the process of creating music is based on clearly defined musical rules, it cannot be completely formalized. To reduce the role of the user in the process of choosing the characteristics of a musical composition, as well as to take into account the emotional component (for example, the emotional state of the user-composer), in this work it is supposed to obtain the characteristics of the composition from an image. In the framework of this work, automation of the process of creating music by means of automated generation of sounds from an image is assumed. In other words, the generation of sounds from an image is the process of converting an image into one or more sequences of notes, with a certain fundamental tone and duration [2].

Now, computer music can be used in many industries: creating music for computer games, advertising and films. Also, the generation of sounds based on image can be applied in the educational process [3]. The development of musical perception in preschool children can be in the form of integrated educational activities, which is based on combinations of similar elements in music and arts (the similarity of their mood, style and genre).

Demographic processes of the last decades, characterized by labor shortage and the growing demands of sociomedical to achieve and provide a decent quality of life dictate the need for effective and accelerated medical and social rehabilitation of patients with critical vascular accident (infarction,
To date, there are no available and universal tools for online monitoring and management of the treatment and rehabilitation process in patients, which leads to the preservation of a very high percentage of complications of treatment measures.

An important characteristic of a composition that corresponds to a specific emotional state of a person is a musical genre: for example, classical or relaxing music is more suitable for a calm state, while energetic music is more suitable for sports. In this regard, it was decided to conduct a research and propose an algorithm for generating musical compositions in various musical genres - classical, rock, jazz, etc.

2. The method

Currently, neural networks are a powerful tool for various tasks related to data processing and analysis, including they are suitable for studying the ways of automating human creativity. For example, neural network DeepDream by Google can search for patterns in images and improve them, while system BachBot can generate compositions in the Bach style.

Naturally, neural networks are great for music generation tasks. However, the musical information is quite extensive; both the relationships between the previous states of the composition and between parallel voices (parts) are important. In addition, it becomes a big problem to train the network on various compositions in order, on the one hand, to have a sufficient training sample to obtain a variety of compositions, and on the other, not to retrain the network in various genres, directions, composers and get an absolutely discordant composition at the output.

![Diagram of the method](image)

**Figure 1.** The top-level method for generation of musical compositions in different genre

Therefore, it was proposed to consistently train the neural network on compositions of different

---

2
authors but of the same genre, save the trained models divided by genres, and then predict the composition with a model trained only on a specific dataset of a specific genre - this will give a more relevant output.

Suppose we got various training models on a dataset of various genres, then the process of generating a new composition at the top level is shown as diagram in Figure 1.

3. Training data collection and classification

In the method described in the previous section, the main task is to correctly train the neural network for compositions with different musical genres. And the most important part of this process is the collection and classification of data for training.

In our case, we will train the neural network using midi files - these are files that carry a top-level description of the sequence of played notes with their durations, instruments, and so on. It is this file format that allows us to most accurately obtain a description of a musical composition that is close to a human (as it would be with an ordinary human description - musical notation).

For these purposes, the The Lakh MIDI dataset was taken [4]. This dataset consists of collection of 176,581 unique MIDI files, 45,129 of which have been matched and aligned to entries in the Million Song Dataset [5]. This dataset provides files that contain in the name of the author and the name of the composition in an almost arbitrary form without any genres description, for example F.FENDER.Wasted days n wasted nights.mid, i_am_the_walrus.mid, or IBelieveICanFly.mid. So, the next step in preparing the training data is to classify the midi files by genre.

First, we used the Spotify Song REST API, which provides the ability to search for a specific song and then retrieve the genre by artist. To do this, we obtain the name of the midi file and with this name we send request to the Spotify API to search the song, for example - `/v1/search?q=N COLE Almost like being in love&type=track&limit=2`. This request will search for a song by the search string and return the search result in JSON format. This response contains some song metadata, like name, images and artist block presented in Figure 2.

![Figure 2](image1.jpg)

**Figure 2.** The response of search for song with using the Spotify API

By using this response we can obtain the artist of the song with using the direct link from `href` field. This response contains some artist metadata, like name and genres block presented in Figure 3.

![Figure 3](image2.jpg)

**Figure 3.** The response of search for song with using the Spotify API

Thus, for two requests, we received a list of genres to which this "raw" composition belongs.
We processed all the compositions from the dataset through this mechanism, some of the compositions were not found in the service, for some there were no genres - after filtering all irrelevant files, we received about 50 thousand unique compositions to which 1,213 unique genres correspond. We saved all results to the PostgreSQL database with quite simple structure shown in Figure 4.

![Figure 4. The song metadata database structure](image)

At the previous stage, we received about 1 thousand unique genres, it is obvious that for predicting compositions this is a very large number of genres, most of which are too narrow. So, the next step is to enlarge all genres to a few simple ones. No suitable API was found for this, so it was done manually. We received about 20 simple genres, the distribution of compositions by simple genres is shown in Table 1.

| Simple genre name     | Count of songs |
|-----------------------|----------------|
| Blues                 | 354            |
| Children music        | 219            |
| Relax                 | 109            |
| Choir                 | 58             |
| Classical             | 1,451          |
| Country               | 680            |
| Dance                 | 450            |
| Disco                 | 466            |
| Drum                  | 7              |
| Electronic            | 727            |
| Folk                  | 385            |
| Hip-hop               | 710            |
| Indie music           | 312            |
| Jazz                  | 473            |
| Latin music           | 408            |
| Pop music             | 10,438         |
| Rap music             | 280            |
| Reggae                | 141            |
| Religious             | 197            |
| Rock                  | 11,464         |
| Soul                  | 407            |
| Soundtrack            | 882            |

So, now we have enough data for training different models by songs with particular genre.

### 4. Description of the used neural network

Recurrent neural network (RNN) has recurrent connections which allow the network to store information related to the inputs. These relationships can be considered similar to memory. RNN is especially useful for the study of sequential data, such as music [6].

In TensorFlow, the repeated connections on the graph are deployed into an equivalent feedforward
Artificial Intelligence and Digital Technologies in Technical Systems II-2021

Journal of Physics: Conference Series

2060 (2021) 012023  doi:10.1088/1742-6596/2060/1/012023

neural network. Then this network is trained using the technique of gradient descent, called backpropagation through time (BPT) [7].

There are a number of ways in which RNN can connect to itself with cyclic compounds. The most common are networks with long-short term memory (LSTM) and gated recurrent units (GRU). In both cases, networks have multiplicative neurons that protect their internal memory from overwriting, allowing neural networks to process longer sequences. In this work, LSTM is used. All recurrent neural networks have the form of a chain of repeating modules. In standard RNNs, this repeating module will have a very simple structure, for example, one layer of tanh. LSTMs also have this chain, but the repeating module has a more complex structure. Instead of having one layer of the neural network, there are four interacting with each other in a special way [8].

The first step in LSTM is to decide what information we are going to throw out of the cell state. This decision is taken by the sigmoid layer. This layer looks at the value of \(h_{t-1}\) output and \(x_t\) input, calculates a value in the range from 0 to 1 for each \(C_{t-1}\) state. If the layer returned 1, this means that this value should be left (remember), if 0 - removed from the state of the cell. For example, in the state of a cell, the characteristics of the current measure can be stored - if the measure is not yet complete, then it is necessary to leave the characteristics in memory, if work is already in progress, then new parameters must be memorized.

The next step is to decide what new information we are going to store in the state of the cell. To do this, firstly the sigmoid layer decides what values we will update. Next, the tanh layer creates a vector of new candidate values, \(C_t\), that can be added to the state.

The next step is to update the old \(C_{t-1}\) state in the new \(C_t\) state. To do this, it is necessary to multiply the old \(C_t\) state, thus deleting the information from the state. Then, it’s necessary to add resulting value and it * \(C_t\). Thus, we get new candidate values, scaled by the update coefficient value of each state value.

At the last step, we need to decide what will output this layer. This output will be based on the state of the cell. First, we pass the input value through the sigmoid layer, which decides which parts of the cell state should be output. Then, we process the state of the cell using tanh (to shift the value between -1 and 1), and multiply it by the output of the sigmoid layer.

The behaviour of a neural network is determined by the set of weights and displacements that each node has. Therefore, for the properly work of neural network we need to configure them to some correct value. First, it is necessary to determine how good or bad any output is according to the input value. This value is called cost. Once the cost is received, we need to use the backpropagation method. In fact, it reduces to calculating the cost gradient relative to the weights (differential of the cost for each weight for each node in each layer), and then it is necessary to use the optimization method to adjust the weights to reduce the cost. In this work, we will use the method of gradient descent [9].

For the training of a neural network, it is proposed to feed a vector that contains the following parts [10]:

- note name: MIDI representation of current note. Used to represent the pitch of a note;
- time when note on;
- time when note off;
- the velocity of the note playback.

5. The system architecture
Proposed system architecture is shown in Figure 5.

6. Conclusion
The result of the work is an analysis of the approach to generating of musical compositions in different style. A method for generating music was proposed and described, a method for collecting and classifying training data was described, a list of simple genres and the distribution of compositions for training in these genres was proposed, and the used neural network was described.
Figure 5. Proposed system architecture

7. Acknowledgments
The reported study was funded by RFBR according to the research project № 20-37-90004

References
[1] Rozaliev V 2013 Methods and Models for Identifying Human Emotions by Recognition Gestures and Motion The 2013 2nd International Symposium on Computer, Communication, Control and Automation 3CA pp 67–71
[2] Xiaoying Wu 2008 A study on image-based music generation. Master’s thesis, Simon Fraser University
[3] Chereshniuk I 2015 Algorithmic composition and its role in modern musical education Art education 3 65-68
[4] Raffel C 2016 Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching Doctoral thesis, Columbia University pp 161-163
[5] Bertin-Mahieux Th, Ellis D, Whitman B and Lamere P 2011 The Million Song Dataset Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011) pp 591-596
[6] Mazurowski L 2012 Computer models for algorithmic music composition Proceedings of the Federated Conference on Computer Science and Information Systems pp 733–737
[7] Doornbusch P 2014 Gerhard Nierhaus: Algorithmic Composition: Paradigms of Automated Music Generation Computer Music Journal 34(3)
[8] Mikolov T 2010 Recurrent neural network based language model Proceedings of Interspeech International Speech Communication Association 9 1045–1048
[9] Kim H, Ao S and Mahyar A 2013 Transactions on Engineering Technologies: Special Issue of the World Congress on Engineering and Computer Science Springer Publishing Company, New York pp. 796
Fernandez J, Vico F 2013 AI Methods in Algorithmic Composition: A Comprehensive Survey Journal of Artificial Intelligence Research 48 513-582