INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2024, VOL. 70, NO. 1, PP. 71-78
Manuscript received November 14, 2023; revised March, 2024. doi: 10.24425/ijet.2024.149516

Sofcomputing approach to melody generation based on harmonic analysis

Jacek Mazurkiewicz

Abstract—**This work aims to create an ANN-based system for a musical improviser. An artificial improviser of "hearing" music will create a melody. The data supplied to the improviser is MIDI-type musical data. This is the harmonic-rhythmic course, the background for improvisation, and the previously made melody notes. The harmonic run is fed into the system as the currently ongoing chord and the time to the next chord, while the supplied few dozen notes performed earlier will indirectly carry information about the entire run and the musical context and style. Improvisation training is carried out to check ANN as a correct-looking musical improvisation device. The improviser generates several hundred notes to be substituted for a looped rhythmic-harmonic waveform and examined for quality.**

Keywords—**music generation; harmonic wave analysis; ANN; musical improviser**

I. Introduction

MUSIC has accompanied human civilization since the beginning of its existence. It has been two-rowed; it has been tried to understand it for centuries. Currently, this state of affairs has not changed much. What humanity has already understood about music is used in technological solutions. What it has yet to discover, it is trying to find. Today, we know how mechanical waves work and what the characteristics of human hearing can be used to make better music. On the other hand, many unsolved issues from music theory, acoustics, and psychoacoustics are still waiting to be explored. One of these is the creative process, the totality of how a piece of music is created in a composer's head. This process seemed to be an elusive entity between the creator's thoughts and the paper and pencil. Modern science, however, has brought us much information about how the human brain works and what form the information present in it, which we call thoughts, reflexes, or impulses, takes. Knowledge of how data is transmitted in the human brain can be used to recreate the creative process [16].

The creation of a work of art certainly has some reconstructive elements. New pieces are created based on heard motifs, and their final shape is subject to dozens of decisions. Some choices remain unconscious and escape the perception of the creator. These unconscious decisions, however, are still describable and understandable because they are still made in the composer's brain. Composing a piece is a process in which the brain actively processes much information provided to it - whether from the creator's consciousness or through external stimuli. In composing, the creator's consciousness is active, and he can think about many things during the process. However, there is a twin process in which more is owed to subconscious processes, external stimuli, or musical intuition. This is "live" composing or improvisation [14].

Improvisation is a peculiar kind of creativity because instead of being on a paper or computer screen, the piece goes almost immediately to the listeners. The improviser has an instrument on which he performs music, and a multitude of auditory and musical stimuli surround him. Auditory stimuli are sounds processed by the mind, while musical stimuli result from sounds and are all rules that pin sound impressions into an analyzable whole. Such complex musical information would include chords, their progression, rhythm, meter, style, form, etc. [17] The improviser in his brain makes a quick, intuitive musical analysis [14]. Its result is an improvisation on his instrument. What takes place in a musician's head can be described as a particular complex system, which, at its input, receives appropriately processed sound and musical data and, at its output, generates further sounds matching the information provided. Today, we already know that this system is a network of neurons with a natural ability to analyze data in a complex way. If, on the other hand, we are familiar with the tool that creates the music, then we can attempt to simulate it. It seems that an artificial neural network can cope with improvisation, and constructing such a network and preparing data for it seems not to be an impossible thing, especially since we can possess most of the necessary information on how to create such a simulation and how to provide it with input data. The answers to some engineering questions related to this issue come to us ad hoc, and some will have to be obtained by appealing to intuition in understanding what music is, selecting specific parameters, data sizes, etc. To create a tool capable of simulating musical improvisation, one needs to improvise at the planning and design stage of such a tool at times as well [4].

The purpose of this work is to create a simulation of a musical improviser. An artificial improviser of "hearing" music will create a melody. The data supplied to the improviser will be musical information, that is, so to speak, a preliminarily "realized" sound already. This will be the harmonic-rhythmic course, the background for improvisation, and the previously made melody notes. The harmonic run will be fed into the system as the currently ongoing chord and the time to the next chord, while the supplied few dozen notes performed earlier will indirectly carry information about the entire run and the musical context and style.

After the program code is implemented, improvisation training will be carried out. For this purpose, several

Jacek Mazurkiewicz is with Wrocław University of Science and Technology, Faculty of Information and Communication Technology, Department of Computer Engineering (e-mail: Jacek.Mazurkiewicz@pwr.edu.pl).

improvisation sessions, i.e., recorded examples, will be prepared to show the neural network a correct-looking musical improvisation. After the training, the network will be tested. The improviser will generate several hundred notes, which will then be substituted for a looped rhythmic-harmonic waveform and examined for quality.

Some limitations on the training examples are assumed - the model musical sequence will be a sequence of adjacent notes and a looped, short harmonic run. The stylistics of different patterns will vary, but in some respects, they will be unified - a bar will contain the same number of rhythmic values.

The harmonic pattern will be the same in the training data, and the testing track - neural networks will not be tested on unknown waveforms. The improviser will take at its input a few dozen previously performed notes, five notes of the currently ongoing chord, and the time left to change it. The rest of the input will be reset to zero when the chord has less than five notes.

II. Music theory

Music theory is a set of principles and laws governing the art. It describes tonal systems, scales, tuning, musical notation, etc. It tries to describe the practical aspects of music, its creation, and its performance. It explains ways of composition, orchestration, improvisation, and production using modern technology [3]. One of the tools of the theory is musical analysis. It involves examining a work based on its sound and score but also broader issues such as the composer's entire oeuvre, the times in which he lived, the circumstances of the composition, etc. Musical inference will be the primary tool used to analyse the results. This implies the need to explain selected concepts of music theory. The need for notation of sounds appeared as early as ancient times. Music was tried to be notated in various ways. The protoplast of modern notation became the system created in the Middle Ages. Initially, it was a single line, and the square notes superimposed on it. It could have been a more precise system. Later, it was improved by introducing more lines and making the pitch more precise. Note notation has evolved over the centuries from its original - very symbolic - forms to today's, which is somehow wholly concretized. Nevertheless, notes today are still an intuitive representation of sound.

In the musical sense, a note is a particular object conventionally symbolizing a specific sound [13]. On the other hand, sound is an inexact concept in this sense. Its pitch, intensity, and duration are only partially deterministic. This is due to the characteristics of human hearing and the lack of need to define musical notation mathematically precisely. This need has emerged only recently with the advancement of digitization. The MIDI standard was then defined, which, in a sense, is also a musical notation [5]. The problem addressed in this paper treats musical notation more determinedly.

Thus, it will be convenient to describe a note as a vector of features, each of which is an actual number or an integer. Such a description results from a certain simplicity attempted during the system design stage. The basic notation considers the enormity of the features and nuances omitted from the application. A violinist performing a sound set by the notation can play it quieter or louder, longer or shorter. Depending on the notation, he will play it with the end of the bow or the middle, as a single sound, or a multitude of fast, "minor" sounds. A sound described in the MIDI standard also has many characteristics defined by events such as dynamics, vibrato, periodic louds and mutes, etc. A note played on an instrument or computer is thus described by multiple values. In the context of the training data prepared for the neural network, only the following will be considered: the moment the sound starts, its pitch, and duration. These features are entirely sufficient for the problem under consideration. For simplicity (and harmonization with the MIDI standard), all values will be limited by specific reasonably selected ranges, which will be discussed later in the paper. The term "note" will be used in this work in the abovementioned context [5].

Notes make up larger structures: motifs that build phrases, which form musical sentences. The theory of musical forms handles the analysis of these structures and their interrelationships. A musical form is a way of organizing musical elements in a work [3]. The term suggests an analogy with a baking mold, which gives raw ingredients a particular shape. The analogy is also apt, given that an example of a musical form is the mazurka, a form stylized from a Polish folk dance. Other examples of musical forms are the classical symphony, fugue, and cantata. When creating a piece of music, the composer consciously decides on its form, which he fills with "ingredients," i.e., structures composed of notes. This filling can vary: the ingredients can be interwoven, layered, or form various shapes and mixtures. The way the elements are arranged in a form is the texture. It is the key to further considerations. Musical form in improvised works is less important, and its analysis needs to be included.

In music theory, texture is present in a musical work in the form of relationships between the various voices of a piece. A voice is a component of a work composed of musical sentences, characterized by the fact that it is separable as a single entity. It can be analyzed as a single musical entity, the beginning and end of which are located in certain places in the work. A voice, for example, can be a part of a single instrument (flet, oboe), a passage played by a single hand, or even a single finger of the pianist. Due to the strict nature of the work, the following definition will be adopted: a voice is a logically ordered collection of notes in a position that does not overlap, i.e., no note sounds simultaneously with any other. A group of voices performing the same function can be called a layer [7].

If there is a single voice in the work, the texture of such a work is called monophonic (one-voice), while if there are more voices, each of which is "equally important" - then the texture is polyphonic. The case in which individual voices perform certain functions that differ from each other is called homophony, and such a piece's texture is homophonic [16]. This is the most common type of texture in contemporary musical works. Improvisation usually proceeds, so the improvising musician plays a specific part to the accompaniment heard. For example, it could be a jazzman playing the saxophone, accompanied by a pianist, bassist, and drummer. In such a case, the improvisation factor is homophonic, as the melody is played on the saxophone, another harmony created by the piano and bass, and another rhythm played by the drummer. Melody, harmony, and rhythm are musical layers, and the fields based on the company's assumptions about the training data. It was decided that the melodic layer would be taken over by an artificial improviser and the harmonic-rhythmic layer by a human trainer. In the following subsections, these will be discussed in detail. Harmony has several meanings in music. It is a science describing the relationship between simultaneously occurring sounds and the very term for the phenomenon of simultaneous sounding [3]. Several notes played at once form a chord, while several chords are a harmonic course. The theory of harmony defines the rules for combining sounds into chords and creating correct harmonic runs. A distinction should be made between harmony, the meaning of general, and harmonics, a specific style found in work (e.g., jazz harmonics, Chopin harmonics) [17].

The music's harmonic-rhythmic (or shorter: harmonic) layer performs the background function, giving the piece a particular shape. As the name suggests, it consists of two inseparable components - harmonics and rhythmics. Rhythmics are all temporal relationships between sounds or groups of sounds. Harmonics, conversely, are the high-bone relationships between sounds and the functional relationships between chords. Thus, this layer's horizontal (time course) and vertical (chord) structures are essential. It is worth mentioning that the features of rhythmics and harmonics significantly affect the nature of other layers. The playing style of a jazz ensemble influences how an improvising saxophonist "feels" the music and how it plays. It's hard to imagine improvising a completely different song than the one currently being played by the band [15].

The harmonic-rhythmic layer was tried to be designed precisely with the music background in mind. It is intended in the training data to be the glue, the foundation on which the higher layer will be built. The design of this layer is simple in its conception. A musical course consisting of several chords was created and looped appropriately as needed. Care was taken to make this course as simple and pleasant as possible.

A melody consists of time-ordered sounds of a particular pitch [8]. The melodic style adopted in a given piece is the melody. In a homophonic texture, the melodic layer contains one voice. By convention, it is a voice with tones higher than the other layers. This layer usually does not have double notes, but its analysis proceeds in time and pitch. Pitch is studied between the individual sounds of a melody and relating them to other layers. This is because melodics strongly connect with the different elements of a given piece. In the case of the melodic layer, it is worth mentioning the range, which is a limited set of sounds that contains notes that match the tonality of the harmonic course. A well-constructed melodic layer exposes the notes matching the system (found in the gamut). This does not mean that sounds coming from outside the range are always wrong - if they are musically justified, they are called extraneous sounds. A well-constructed melodic layer is based on the sounds of the range and uses outside sounds to enrich the melody line [1].

The human brain can naturally analyze sounds from a musical angle. In the operation of this natural tool, one can see specific patterns that contribute to how we hear music. One of the basic things we are sensitive to is harmonic sound. It consists of so-called fundamental tones. Each tone is a sine wave with a frequency of multiple of the lowest tone. The second thing is rhythmic, or in general, any simple or more complicated repeating temporal patterns in musical structures. The human brain is also sensitive to more abstract musical concepts, such as layer division, fragmentation, compositional style, harmonics, spatial placement of sound, instrument timbres, etc. The field that deals with the brain's perception of music is psychoacoustics. Some solutions in the improviser's model can be explained by discussing psychoacoustic issues.

One of them, for example, is music perception, the natural ability to stay in a musical context. In practice, it is the phenomenon that the piece of music heard is understood by people at the level of structures and so-called figures rather than the mere figurative properties of sounds. A figure, as understood by music psychology, is a specific entity governed by certain organizational principles, in which established relationships between elements allow abstracting the specific values represented by these elements [9]. As a result, the sounding of, for example, two notes following each other creates a clear relationship between them, and both are considered together. One chord is perceived as the foundation. However, the same chord preceded by some other chord can become a chord that discharges or creates musical tension. A pure, "simple" chord can become an outlier in a piece when it is among the chords of the "complicated."

Thus, depending on the context, sound, chord, and note can be understood differently. Music is something undeniably felt by the human brain. Musical relations in a piece of music are not just an invention of music theorists. Music has its principles that can be comprehended. Still, it is optional to have the ability to verbalize and understand them to hear and understand musical pieces in an almost identical way [11]. Despite the lack of musical knowledge, people can distinguish between typical musical sequences (i.e., those that "conform" to the rules) and those that are less common. It is this understanding of the musical context that the improviser model sought to implement.

III. Training Data

The concept of an artificial improviser is an extensive issue, so several assumptions must be made at the design stage to focus on the most critical issues. For an improviser to start making music, it must have some foundation, background, and accompaniment. To do this, a harmonic and rhythmic layer must be created, including a looped harmonic progression. This is a simple yet natural musical performance - the ensemble plays a riff or a repeating sequence. The improvising musician will, after some time, anticipate the succession of chords. Such a prepared layer is only one of the two necessary elements. After all, it is still required to craft a pattern, indicating to the improviser "in practice" style, and certain principles are hidden among the notes provided to him. Therefore, you must make a melodic layer, a pattern improvisation. It should be clear and readable enough for an artificial neural network to cope with understanding the general rules and laws. It can be predicted that the melodic layer created by the improviser should have almost the same melodics (musical style). It should manifest in relatively similar melodic phrases, melodic line leading (i.e., melody rising and falling), notes used, and note lengths. An interesting study that can be prepared under such conditions is training on several stylistically different melodic layers. The idea is to prepare several harmonic runs and record a separate pattern for each. In theory, the stylistics of the generated improvisation should vary and, in each case, resemble the pattern's stylistics [14].

An improviser should have "innate" human-like abilities. The realization of the ability to understand musical context can take place in many ways. In the present work, it was decided to prepare the improviser so that when performing a given note, he will have dozens of previous notes in his memory, which should, in theory, allow him to take more care of the coherence of the whole improvisation. In addition, the notes that make up the currently sounding chord and the time after which it will be changed to the next chord will remain in the network's memory. They sensed this "music" for the harmonic-rhythmic layer. The system performs a preset number of notes, with each note still played in the improviser's "mind" for some time. The chord also remains in memory, as well as its remaining duration. All of this can affect the note just being created. This attempts to map the phenomena occurring in the human brain, generally described as "feeling" music [6].

The input point of the designed system was music data saved in the MIDI format [5]. For their initial processing, the music21 library was used [18]. With the help of this library, it was possible to extract the necessary data without going into the implementation details of the format itself. During processing, irrelevant data was filtered out, and individual MIDI tracks were sorted in memory [1]. After filtering, it was possible to iterate over the strings of sounds that make up the song. Data must represent a sequence [10].

A single training packet contains two lists - an input list and an expected output list. The input list is created from the notes, the chord currently in progress, and the time left until it's sounding. In practice, dozens of previous notes were placed in the input, but the creation of training packets can be discussed using a simplified example. So, let there be a sequence of notes and chords (Fig. 1).

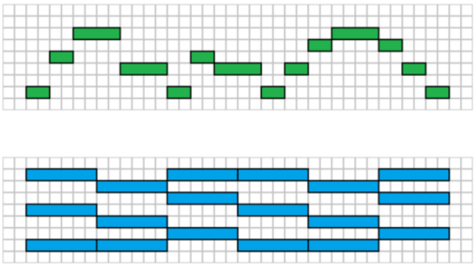

Fig. 1. Example musical sequence: green notes are the model melodic layer, and the blue notes are the harmonic waveform

The packet will include the seven green notes of the melodic line, three notes of the non-beat chord, and the time remaining for the chord change, indicated by the red line, covering four bars in this example. The highlighted eighth note from the sequence will be expected in the output (Fig. 2).

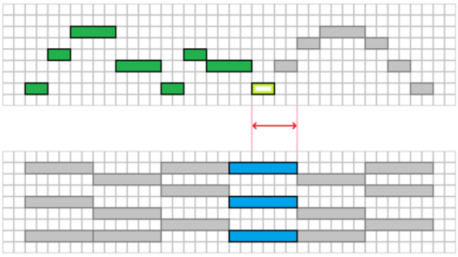

Fig. 2. Elements included in the training package

A simplified model of the system is shown in Fig. 3. At its input, the data is normalized. Each note is represented by duration (*t*) and pitch (*p*). For seven notes of a melody and three notes of a chord, the vector has a size of 21. At the output of the improviser, the following note located in the set sequence is expected. At the very beginning, the number of packets is determined. The main loop iterates over the subsequent notes in the manner discussed above. The harmonic sequence is, of course, looped the appropriate number of times. The logical condition that checks the length of the list with ready packages is created so that if the number of notes is too small, the test package will be made anyway. The method can prepare a cull packet with some items in the list zeroed out with such a procedure. Such a packet can be the initial sequence when testing the network. The numbers processed in the method are pre-normalized at this stage. The pitch is divided by 100 and the duration by 300. This, in effect, reduces the numbers while preserving their logic and structure. The numbers 100 and 300 were chosen by trial and error and are reasonable ranges around which the features of the notes oscillate. The note heights of the model melodies usually come from the range of 40 - 80, which will give a range of 0.4 to 0.8 after normalization. Their average duration was examined by parsing the melodies and manually checking the values. It turned out that the unit of measure in the chosen format had a length of 96. After these steps, the number 300 was chosen as the denominator in dividing the duration of a note.

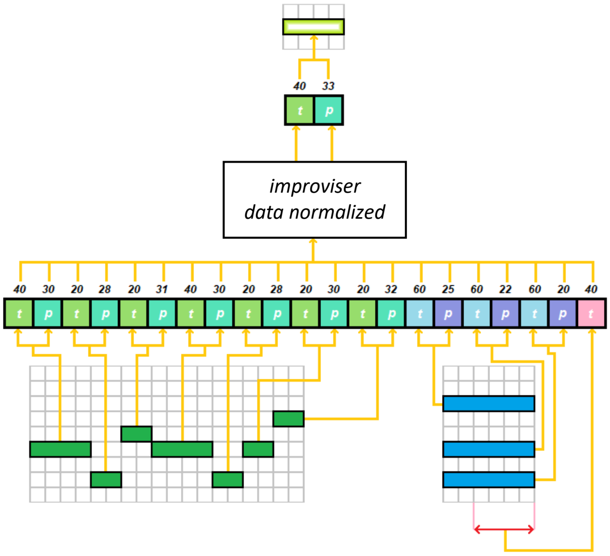

Fig. 3. Diagram of inputs and outputs entering the system

The prepared packages are created in the method in which the neural network is trained. Thanks to the separation and encapsulation of the technique's functionality implemented higher up in the architecture, they are smaller and more readable, and instances of training packages can serve in many places in the code. During the study, two musical sequences differing in style were prepared. One had 16038 notes and was styled as pop music, while the other had 2081 notes and was in orchestral style. So one was very "rustic" and fast, and the other was slow and calm. This was to contrast the training data and see if the stylistics of the generated pieces remained in harmony with the style of training sequences. The most frequently tested case was when the neural received 55 notes per input, meaning there were 15983 training packets per training epoch for the first sequence and 2026 for the second.

IV. Neural Improviser Topology

Neural networks - the improviser's brain - are created using the TensorFlow library [12]. Networks differ in the number of hidden layers, the number of neurons in the layers, and the types of activation functions. The number of hidden layers in different models varies from one to eight (Table I)..

TABLE I
TOPOLOGIES OF THE TESTED NEURAL NETWORKS

Id	Number of hidden layers	Number of neurons in the following hidden layers	Activation function
1	1	3	relu
2	1	input_layer_size+30	relu
3	2	4, 4	relu
4	1	30	relu
5	8	16, 13, 7, 6, 7, 13, 16, 18	relu
6	1	16	tanh
7	2	14, 4	tanh
8	2	14, 10	tanh

The Adam optimizer is used, the learning rate is 0.001, and the momentum is 0.9 [2]. Training is done in so-called epochs. Each epoch represents one whole training data show. The number of epochs is set as a parameter. Networks have been trained in many ways, mainly using a single optimizer. The most frequently changed parameter in this process was the number of epochs. Small networks were given a value of about a few thousand; for the largest, it was up to a hundred thousand. Testing the network is understood as generating an entirely new melodic layer from very little initial data. The neural network gets a harmonic sequence, to which it previously learned to improvise, and from a few to a few dozen notes, representing the beginning of a new improvisation. The artificial improviser is a collection of different neural networks. Each presented the same problem: musical improvisation in a given style. Knowing the models of the networks and the results they generate, one can try to put them together [18].

V. Results

In the case of the networks Id 2 and Id 5 (Table 1), the results are unmusical (the sounds possess the shortest possible length) or even impossible to generate. In further compilations, these numbers are omitted. The overdubbed and under-dubbed nets – Id: 4, 7, 8 (Table 1) - were left because their melodies are analyzable, even though, from a musical point of view, they are of inferior quality.

The second approach to studying the results is musical analysis, which can be done on the melodies created. The results of each neural network will be discussed separately and in the context of other melodies. It is worth recalling here that two utterly different test sequences were created. The first had 16038 notes and was kept in the style of popular music, while the second had 2081 notes and an orchestral style. The "entertainment" sequence had faster notes, changing more often and giving the impression of much movement, while the second was quieter and slower.

The first two neural nets - Id 1 and 3 - created melodies for the first sequence. Even though many sounds went out of range, the general style was maintained. Despite the small number of neurons in the hidden layer, the first network did quite well in reproducing the general idea; nevertheless, from an overall angle, it performs worse than the third network. The melody it generates has too much ambitus (distance between the highest and lowest notes), which can be heard immediately at the beginning of the sequence.

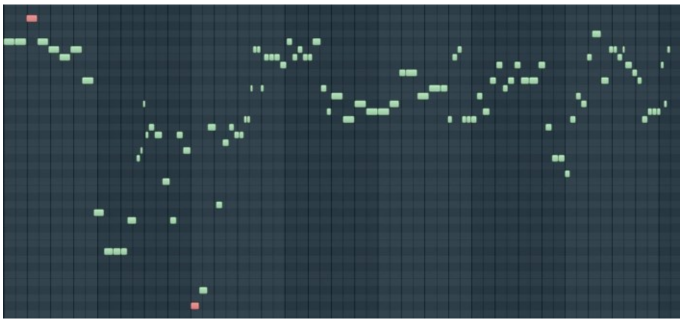

Fig. 4. Melody of net Id 1- the highest and lowest tones highlighted in red

In Fig. 4, you can see quite a bit of chaos in the melody leading, which contrasts with the model melody (Fig. 5).

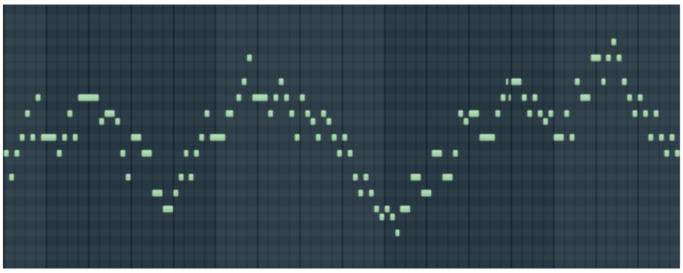

Fig. 5. The first pattern melody (excerpt)

The sequence of net Id 1 cannot be considered complete randomness. Instead, it is a clumsy attempt to replicate the idea behind the pattern. The melody completely misses the sounds, and the rhythmics are very simple, missing the rhythmics of the harmonic layer. Instead, the sequence of net Id 3 looks different (Fig. 6).

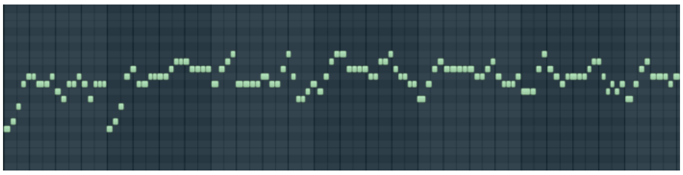

Fig. 6. Network Melody Id 3 (excerpt)

The above sequence is closer to the template regarding how the melody is provided. Although improviser Id 3 limited his range of sounds below that presented in the template sequence, he nevertheless evidently made more of an effort to mimic the up and down movements of the sounds. The rhythmics of this melody are more orderly. It still needs longer runs with sounds that follow the range, but occasional short moments of several minutes sound good. The pattern sequence is a man-made improvisation. A natural occurrence in such improvisation will be the use of sounds that come directly from the chords of the accompaniment, especially at the beginning of the harmonic sequence. In other words, the base sound of a chord will often repeat at the beginning of each repetition of the harmonic waveform (Fig. 7).

The white lines (Fig. 7) separate the sections where the harmonic sequence repeats. The white arrows point downward to the first sounds of the series. As you can see, the pitch of these sounds is the same (the last sound lies 12 semitones higher, so in a musical sense, it is the same as the first three sounds). Neural network Id 3 understood this principle presented in the training data and tried to implement it in its composition (Fig. 8).

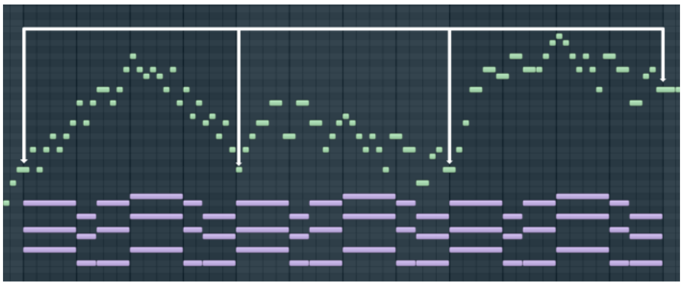

Fig. 7. Pattern melody with labeled repeating sounds

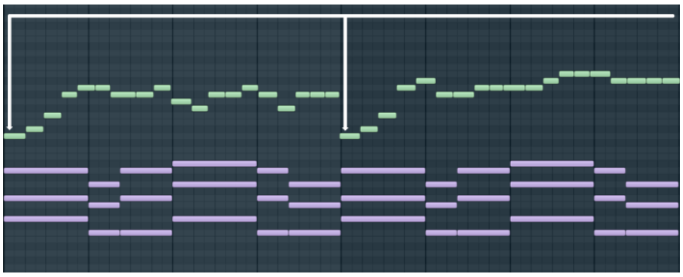

Fig. 8. Network melody Id 3 with labeled repeating sounds

The marked sound is precisely the same sound used during the improvisation. This indicates a higher understanding of the idea of improvisation than in the case of net Id 1. In addition, the initial fragments of the melody for subsequent bars indicate a specific overall concept of the melody (Fig. 9).

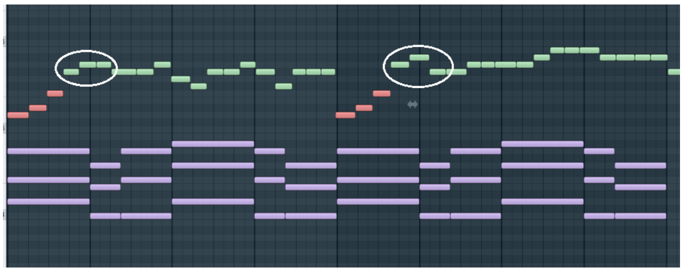

Fig. 9. Melody of network Id 3 with marked beginnings of successive repetitions of the harmonic sequence

The notes marked in red are identical, and those in the white circle are an interesting attempt to develop the melody - the first sequence begins slightly lower. In the following sequence, the melody goes upward, building musical tension. Two networks with different topologies have formed two qualitatively different tunes. The third network is far superior to the first. The melodic sequence of the improviser Id 1 is much more random, as if the network needs to understand what it is supposed to do thoroughly. It is also worth mentioning that a correlation was noted between the number of layers of a network and the quality of its solution - networks with fewer layers produced songs that were simpler and often completely wrong.

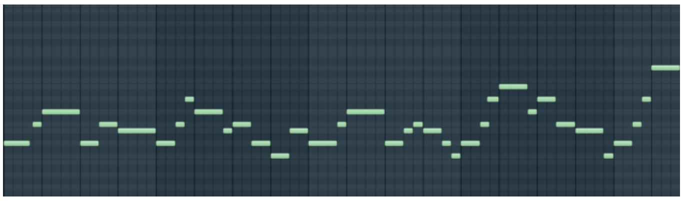

Fig. 10. Second master melody (excerpt)

For networks with more layers, the music seemed to have a deeper meaning and a better thought-out structure - as seen in the above comparison of networks with one and two small layers. However, this principle was only sometimes fulfilled. Improvisers numbered Id: 4, 6, 7, and 8 created melodies under the second sequence (Fig. 10). Network Id 4, marked as overtrained, generated a tune closest to complete randomness. It's hard to say why the network with thirty neurons did so well and not differently. From another perspective, compared to network Id 2, due to the erroneous result of the absent one in this list, network Id 4 does not fare so severely since its melody can be seen and heard (Fig. 11).

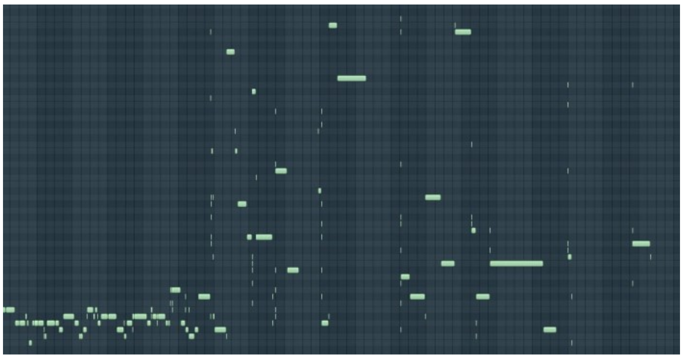

Fig. 11. Network Melody Id 4 (excerpt)

Interestingly, the network creates a strange melody only from a particular passage. This may be because, for the first few dozen iterations, the neural network's memory contains notes that begin its improvisation. This may mean so much that the network, in the case of known data (for such was used as this "beginning"), more or less understands what it should generate; when these model notes begin to disappear from memory, then the improviser gets lost. The conclusion that can be drawn from the analysis of this melody is that the network is "too smart" because it has learned the set sequences by heart, and when encountering new ones, it stops coping. Hence, this is why it is labeled as overtrained. In addition, comparing this network with network Id 2, one can see that 30 neurons are still enough to generate anything, but 140 is too many. The best musical created during the study, the improviser melody numbered Id 8, was generated after the first training. The network used only three pitches of notes, fewer than were recorded in the model sequence, but they were nevertheless developed musically correctly. Unfortunately, one of the sounds went off the scale and was used 22 times (hence the 11% error). If it had been a tone a semitone lower, the melody of net Id 8 could have been considered a complete success (Fig. 12).

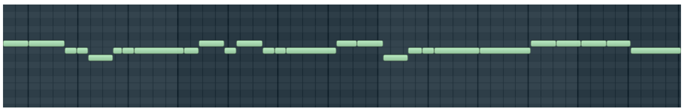

Fig. 12. Network melody Id 8 generated after the first training (excerpt)

The leading of the above melody is thoughtful, as each sound fits in and merges into the currently sounding chord of the sequence. The melody, despite being very minimalistic, sounds good. The line is conducted in the style of a pattern. Like network Id 8, network Id 6 created a tune that still fits the pattern but needs to be corrected, as it is more chaotic and conceptually inferior. This is analogous to the situation with networks Id 1 and Id 3. Both improvisor Id 3 and Id 8 have two layers each. Id 3 is slightly smaller. On the other hand, Id 1 and 6 have one layer each, and Id 1 is smaller. The Id 6 and Id 8 nets are larger versions of Id 1 and Id 3. Intuitively, comparing them should yield similar conclusions, and indeed they do. Here's a snippet of the network's melody Id 6 (Fig. 13).

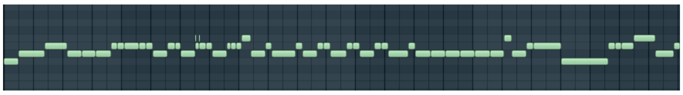

Fig. 13. Network melody Id 6 (excerpt)

It is, in fact, similar to the net sequence Id 8; nevertheless, musically, it is much worse. The rhythm of this melody is disturbed; some short sounds (visible in the picture) need to be corrected in the melodic sequence. Network Id 6 used more sounds and wrong ones, resulting in a musically inferior piece.

Both networks were labeled as under-trained because, looking at their melodies, they needed to understand the idea of the training data presented. At first glance, the sequences may appear identical. Nevertheless, several differences make network Id 8 outclass the solution of network Id 7. The first issue is rhythmicity. It is the same note repeated two hundred times. The melody of the improviser Id 8 is much more interesting because the rhythmics consist of several "classes" of sounds, each of which has a different length. There are short, medium, and long sounds, but none are extreme. The whole sounds exciting and, surprisingly, does not give the impression of a bland melody.

The second thing is the pitch of the note used (F5 and lower by half a tone E5). Although the sounds used by both grids belong to the scale (hence the 0% error), the note that improviser Id 7 decided on does not fit all chords equally. Although this sound is a component of the first, most essential chords, it creates an unintuitive clash with the second and sixth chords. Net Id 8 avoided this problem by using a sound a semitone lower, resulting in fascinating, complex relationships between the chords. The sound of the seventh network creates consonances with the chords that are atypical of the standard sequence. At the same time, the note of improviser Id 8 resonates with the harmonics in a more classical yet non-trivial way.

The network also used one other sound near the end of its sequence, which leads us to suspect that it did not loop in its creative process since, in creating this sound, it only had its notes made earlier already in mind. This sound also comes from the scale and is the most essential sound of the first chord.

CONCLUSION

Machine learning projects have it that it takes work to predict research results. Waiting several days for results that may be completely wrong is inherent in this field and cannot be avoided. In the case of the present project, developing these long waits is satisfactory. The musical quality of the generated samples may be a debatable issue; nevertheless, from the point of view of this work, the critical fact is that the analyzable results carry much valuable information about the operation of neural networks, their topology, and learning methods. The lack of musically correct results can also be considered as a result because the enormity of the work that the neural networks had to do is not insignificant, and it contains a lot of valuable hints about things that may prove helpful in case of a possible attempt to develop this work with additional, unrelated threads.

In terms of music, there is a correlation between the training data presented and the music generated during testing. Musical analysis shows that the improviser attempted to create a melodic part using the general style of the given training data. This relationship can be seen especially when one compares samples produced by similar networks after training with different master sequences (as in the case of networks Id: 1, 6, 3, and 8). This is a success of the present study. One can see much more when considering different topologies of neural networks and looking at their results. The "shorter" networks, with one or two hidden layers, generated compositions generally inferior in terms of overall compositional conception. This manifested in the fact that such a piece was, in a sense, a collection of random notes between which no idea was contained, such as the rise and fall of the melody, climaxes, etc. Of course, a random set of notes is not a completely random set because, in general, the notes present in songs with a weak musical concept still fit into the time and pitch range of the notes derived from the test data. The pieces, therefore, sounded like the improvisation of someone who is entirely ignorant of the principles of music but has a sense of it and happened to sit down at an instrument whose principles he also does not understand.

Networks with more hidden layers worked in two ways. Some of them created songs much better in the context of the overall structure, as manifested by the rise above and fall of melodies, certain melodic figures made along those in the training data. There were still many misfits in the correct range of sounds, but the network understood the idea of running them. Sometimes, however, the concept of musicianship and improvisation was utterly foreign. Songs generated by such networks had, for example, pitch different sounds, but each duration was zero. Once it happened that the duration was non-zero, it was most often constant, at which point all sounds had a single pitch. This is a classic effect of overtraining the network or selecting the wrong topology for the problem presented.

The most puzzling neural network was network Id 8, which generated a song that used two sounds, one of which appeared 199 times and the other only once. The rhythmic pattern was not uniform, as one could hear a rhythm that paired with the structure of the training data. Interestingly, the sound that was used 199 times was a sound that matched each of the chords performed. It's hard to determine whether this network was overlearned or discovered some exciting way to average the training data and generate music that fit into the overall harmony of the piece in the most minimalist way it could.

The results discussed are so interesting, and the potential is so untapped that there are still many areas that could be touched upon in further research on the nines. The original goal of the project, for example, was to train improvisers in such a way that they could make music to previously unknown sequences. Besides, only one type of network - feedforward - was used, and the parameters changed were mainly the topologies of the models. The effect of learning rate, or momentum, on the results could be investigated. Many more activation functions could be used. Various musical sequences could be created, and an even deeper analysis could be made of how artificial intelligence behaves when interacting with it. Even the results presented in this paper could hide many more interesting structures and relationships. The topic raised is undoubtedly amenable to further analysis, and although it is explored in such detail, it still needs to be more robust. The possibility of further investigation remains open.

REFERENCES

[1] J. P. Briot, F. Pachet, “Deep learning for music generation: challenges and directions”, Neural Comput. Appl., vol. 32, no. 4, pp. 981–993, 2020.

[2] J. P. Briot, G. Hadjeres, F. D. Pachet, “Deep Learning Techniques for Music Generation”, Springer Nature Switzerland AG, 2020.

[3] D. Herremans, C. H. Chuan, E. Chew, “A functional taxonomy of music generation systems”, ACM Comput. Surv. (CSUR), vol. 50, no. 5, pp. 1–30, 2017.

[4] C. F. Huang, C. Y. Huang, “Emotion-based AI music generation system with CVAE-GAN”, in 2020 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), pp. 220–222, 2020.

[5] T. M. Association, “Standard MIDI Files (SMF) specification”, https://www.midi.org/specifications-old/item/standard-midi-files-smf, 2020.

[6] Y. Bengio, P. Simard, P. Frasconi, “Learning long-term dependencies with gradient descent is difficult”, IEEE Trans. Neural Networks, vol. 5, no. 2, pp. 157–166, 1994.

[7] K. Zhao, S. Li, J. Cai, H. Wang, J. Wang, “An emotional symbolic music generation system based on LSTM networks”, in: 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), pp. 2039–2043, 2019.

[8] A. Karpathy, “The unreasonable effectiveness of recurrent neural network”, https://karpathy.github.io/2015/05/21/rnn-effectiveness/, 2015.

[9] H. G. Zimmermann, R. Neuneier, “Neural network architectures for the modeling of dynamical systems”, in: A Field Guide to Dynamical Recurrent Networks, pp. 311–350, IEEE Press, Los Alamitos, 2001.

[10] S. Mangal, R. Modak, P. Joshi, “LSTM based music generation system”, arXiv doi:10.17148/IARJSET.2019.6508, 2019.

[11] S. Hochreiter, J. Schmidhuber, “Long short-term memory”, Neural Comput. vol. 9, no. 8, pp. 1735–1780, 1997.

[12] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, J. Schmidhuber, “LSTM: a search space odyssey”, IEEE Trans. Neural Netw. Learn. Syst., vol. 28, pp. 2222–2232, 2017.

[13] A. Everest, K. Pohlmann, “Master Handbook of Acoustics”, 5th ed. edition, New York: McGraw-Hill, 2009.

[14] B. Thom, "Unsupervised Learning and Interactive Jazz/Blues Improvisation," in American Association for Artificial Intelligence, 2000.

[15] I. Simon, D. Morris, and S. Basu, "Exposing Parameters of a Trained Dynamic Model for Interactive Music Creation," in Association for the Advancement of Artificial Intelligence, 2008.

[16] C. Schmidt-Jones, “Understanding Basic Music Theory”, Rice University, Houston, Texas: Connexions, 2007.

[17] P. Ponce, J. Inesta, “Feature-Driven Recognition of Music Styles”, Lecture Notes in Computer Science 2652, pp. 773–781, 2003. htdoi:10.1007/978-3-540-44871-6_90

[18] J. Mazurkiewicz, “Softcomputing Approach to Music Generation”, in: Dependable Computer Systems and Networks. DepCoS-RELCOMEX 2023. Lecture Notes in Networks and Systems, vol 737, pp. 149–161, Springer, Cham, 2023. https://doi.org/10.1007/978-3-031-37720-4_14