

Adoption of Artificial Neural Network for Rest, Enhanced Postprocessing of Beats, and Initial Melody Processing for Automatic Composition System

Kyunghwan Kim*, Sung Hoon Jung**

Abstract

This paper proposes a new method to improve the three problems of existing automatic composition method using artificial neural networks. The first problem is that the existing beat post-processing to fit into music theories could not handle all the cases of occurring. The second one is that the pitch space generated by artificial neural networks is distorted because the rest is trained with the pitch on the same neural network with large values. The last problem is caused by the difference between the initial melody and beats given by user and those generated by an artificial neural network in the process of new composition. In order to treat these problems, we propose an enhanced post-processing of beats, initial melody processing, and adoption of artificial neural network for rest. It was found from experiments that the proposed methods totally resolved the three problems.

Keywords : Automatic Composition, Artificial Neural Networks, Beat Postprocessing, Initial Melody Postprocessing

자동작곡시스템에서 쉼표용 인공신경망 도입 및 개선된 박자후처리와 초기멜로디 처리

김경환*, 정성훈**

요약

본 논문에서는 기존의 인공신경망을 이용한 자동작곡 방법에서 발생한 세 가지 문제점을 개선하는 새로운 방법을 제안한다. 첫 번째 문제는 인공신경망이 출력한 곡의 박자를 음악이론에 맞도록 후처리하는 것에서 모든 경우를 처리하지 못하여 발생한 문제이다. 두 번째 문제는 음표를 학습하는 인공신경망에 음표와 구분되는 큰 값으로 쉼표를 같이 학습하다보니 음표공간이 왜곡되어 발생하는 문제이다. 마지막 문제는 새로운 곡 작곡 시 사용자가 작곡해서 넣어준 초기 멜로디와 박자가 인공신경망이 출력하는 나머지 멜로디와 박자와 어울리지 못하여 발생하는 문제이다. 본 논문에서는 이러한 문제를 해결하기 위하여 개선된 박자 후처리 알고리즘과 초기 멜로디 처리 방법을 제안하였으며 쉼표용 인공신경망을 새로이 도입하였다. 실험결과 새로 제안한 방법이 기존의 방법에서 발생한 세 가지 문제점을 모두 해결하는 것으로 판명되었다.

키워드 : 자동작곡, 인공신경망, 박자후처리, 초기멜로디 처리

1. Introduction

※ Corresponding Author : Sung Hoon Jung

Received : October 21, 2016

Revised : December 03, 2016

Accepted : December 20, 2016

* Electronics and Information Engineering

Tel: +82-2-760-4344 , Fax: +82-2-760-4435

email: shjung@hansung.ac.kr

** Hansung University

▣ This research was financially supported by Hansung University

Recently, convergence with information technology (IT) has been taking place in various fields. Typically, there are NIT that fuses with nanotechnology (NT) and BIT that fuses with biotechnology (BT). Google has recently tried to adopt information technology in the field of culture and art using the Alphago that won the match against Sedol Lee. The Culture and art has been regarded as a unique area of human beings to date. Attempts to integrate information technology, especially artificial intelligence technology, into cultural technology (CT) will continue in many different directions. CIT technology, which is a fusion of culture technology and information technology, will greatly promote Korea's cultural prosperity. Currently, the Korean wave is spreading in Asia and Europe centered on K-POP. The spread of Korean wave is increasing the economic benefits such as the export of Korean products or the increase of tourists. Therefore, we have been studying automatic composition that automatically generates music using artificial intelligence technology as a fusion of culture technology and information technology [1,2].

As humans listen to existing songs and compose new songs, artificial neural networks that imitate the human brain have been used to learn existing songs and create new ones using the learned results [1]. However, various problems occurred in the process of composing a new song in the existing researches. First, if a music piece generated by using artificial neural network does not fit the music theory in terms of beat, it is solved by beat post processing, but there is a problem that post processing can not be performed in all cases. In this paper, we propose a post-processing algorithm that can be processed in all cases.

Another problem in the existing automatic composition method is that the notes and rest are learned together in one neural network. In order to distinguish the rest from the note, we

decided to set it to a large value, but in the end, this method distorted the output melody space. To solve this problem, we introduce a new rest neural network and propose a method to learn the notes and rest separately.

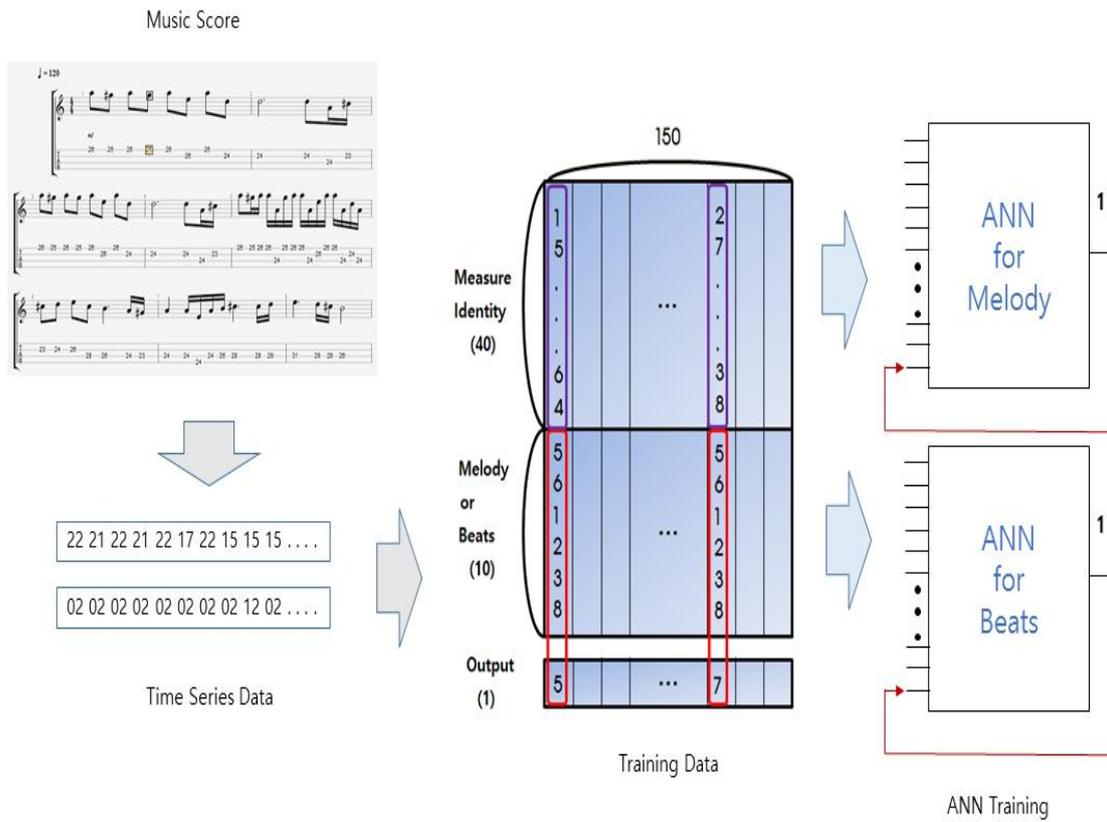
The final problem in the existing automatic composition is that the initial melody and beat that puts in order to make a new song after artificial neural network learning does not match the melody and beat generated by the trained artificial neural network. In order to solve this problem, we propose a method to randomly generate initial melodies and beats of artificial neural networks and not use them as a result of actual composition. Various experiments were carried out by applying the newly proposed method. Experimental results show that the existing problems are solved and more natural music is output.

This paper is organized as follows. Section 2 introduces the existing automatic composition method. Section 3 describes three problems and solutions for the conventional automatic composition method. Section 4 explains the experimental results and this paper concludes with the Chapter 5.

2. Existing automatic composition method

2.1 Introduction to Automatic Composition

Computer-generated automatic composing is called algorithmic composition and has been studied for a long time [3~13]. The most used artificial intelligence techniques for automatic composition are evolutionary algorithms that evolve group individuals and make them more and more good individuals [3-6]. In particular, genetic programming is very useful because it can represent music in a hierarchical manner from a note to a measure and from a measure



(Figure 1) Overall Structure of Automatic Composition System

to a syllable [3]. However, the biggest problem of the method using evolutionary algorithms is that it is difficult to evaluate the fitness how well the individuals fit [12]. This is because it takes a lot of effort and time for a person to evaluate short measures expressed as individuals. It is also very difficult to objectively evaluate short measures with a consistent standard. Because of this difficulty, a method of automatic evaluation using artificial neural network has been developed. However, two new problems have arisen that the artificial neural network for evaluation should be trained and the evaluation with the artificial neural network is inaccurate. It is difficult to produce high-quality music with inaccurate evaluation because the evaluation greatly influences the evolution of the music.

The second most used artificial intelligence techniques are machine learning, typically artificial neural networks[7~11]. Because artificial neural network has learning function, it can learn and output music. Since music changes with time, we can use recurrent neural networks that can learn time series data or feedforward neural networks by representing song data recursively [12]. After learning one or more songs in the artificial neural network, the composition is performed by outputting new songs. However, because the song has many repetitive parts, it is difficult to learn and it takes a long time to learn. Other methods for automatic composition include the use of Markov chains, and the use of finite state machines [12].

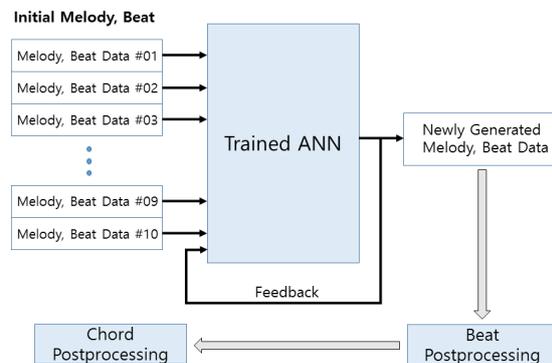
2.2 Previous methods

We performed automatic composition research using artificial neural networks, not evolutionary algorithms since it is very difficult to evaluate individuals [1,2]. In this study, we constructed a recursive data of training songs and trained them to the feedforward neural networks by error back-propagation algorithms like existing methods [1]. Two artificial neural networks are used: one is for training pitches and rests and the other is for training beats. In order to learn the music, the musical pitches, rests, and beats are represented by time series numerical data. We used 2⁴ octaves, which are often used among 7 octaves in music. Since there are total 12 pitches in an octave such as C, C#, D, D#, E, F, F#, G, G#, A, A#, B, we represented all pitches as numbers from 1 to 36. In the case of rests, we set them a large value of 50 in order to distinguish the pitches from 1 to 36 [1,2]. In the process of composition, the values from 1 to 36 are interpreted as pitches and the values larger than 36 are interpreted as rests. For beats, we assumed that the 16th note is 1 in the 4/4 beat and allowed only one dot, so total eights beats such as 1, 2, 3, 4, 6, 8, 12, and 16 were used[2].

In order to learn an existing song, song information should be converted into recursive learning data. The constructing method of the recursive learning data consists of inputting 10 notes and 10 beats to be learned and outputting the 11th note and beat when using 10 inputs of artificial neural network [1]. The next learning data is obtained by removing the first note and the beat and by inputting from the 2nd to the 11th note and the beat, and then by outputting the 12th note and the beat [1]. All training data is composed like this. Our method differs from the existing ones in that it made learning data by adding segment division to the song so that learning

can be done well even if there are many repeated parts. In this case, even if there is a repeated part of the song, the segment division of the training data is different, so that the learning can be smoothly performed. (Figure 1) shows the overall process of the method we studied. Segment divisions are composed of 40 random numbers with the same values of pitches and beats (1 to 36 for pitches and 1 to 16 for beats). For more details, see the papers [1,2].

The training was done using two artificial neural networks for pitches/rests and beats. After training, if 10 different initial melodies and beats are inputted, then the trained artificial neural network generates the remaining melodies and beats. However, the melodies and beats generated from the artificial neural network are only the mapping values of learned melody and beat space, they often do not fit the musical theory. In the case of the beat, incomplete measure occurs, and in the case of melody, the strange harmony between the measures occurs. In order to solve this problem, we employed beat postprocessing and harmony postprocessing [2].



(Figure 2) Beat Postprocessing and Chord Postprocessing

(Figure 2) shows generating a new song from a trained artificial neural network and

postprocessing of beat and harmony. In the previous study, the beat postprocessing was implemented by subtracting the over number of beats from the largest beat when total beats in a measure exceeded 16. For example, in a song of 4/4, the total beat of a measure should be 16. If 4, 6, and 8 beat data are found, then $4 + 6 + 8 = 18$ is exceeded by 2 than 16. In this case, the largest beat, 8, is subtracted by 2 to make 4, 6, 6 by beat postprocessing. However, this method has a problem that it can not be coped with in all cases. The postprocessing of harmony is described in detail in [2] and is not closely related to the subject in this paper, so detailed explanation is omitted.

3. Problems and solutions of existing automatic composition

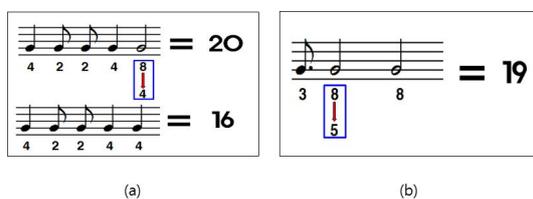
3.1 Beat postprocessing problem

The first problem in our previous study using artificial neural networks is the beat postprocessing problem. We explain this based on the 4/4 beat using the 16th note. The 16th note uses whole note, half note divided whole note by 2, quarter note divided whole note by 4, 8th note divided whole note by 8, and 16th note divided whole note by 16. As a result, when the whole note is set to 1, the beats of 1/2, 1/4, 1/8 and 1/16 are determined. Since

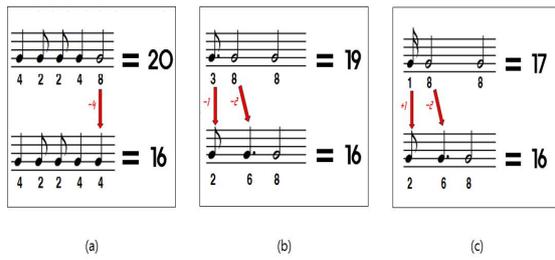
the length of the actual beat is based on the quarter note, the actual beats become to 4, 2, 1, 1/2, and 1/4 from whole note to 16th note. The 4/4 beat means that four quarter notes are in one measure. That is, the sum of the lengths of the beats per one measure becomes four beats. We set the 16th note to 1 to express the beat as an integer. Therefore, the integers of beats from the 16th note to whole note become 1, 2, 4, 8, 16. Additionally, we use one dot note that adds the one-half beat of the note with the dot. Although there are two dotted notes, we don't use them because they are rarely used in general pop songs. Finally, we use the 8 number of integers for beats, i.e., 1, 2, 3(2+1), 4, 6(4+2), 8, 12, 16. The sum of the integers constituting a single measure shall be a total of 16 for the 4/4 beat. To do this, the beat integers are added in order. If the total exceeds 16, then they are regarded as one measure after making the total to 16. We call this beat postprocessing.

In the previous beat postprocessing, we have implemented the method of subtracting the number of beats exceeding 16 from the largest number of beats, as described in Section 2.2 above. However, this method did not work properly in some cases. As shown in (Figure 3(a)), the total sum becomes 20 when the beats are 4, 2, 2, 4, and 8 but it can be a complete measure, 4, 2, 2, 4, and 4 after subtracting 4 from the largest number of 8. However, as shown in (Figure 3(b)), the total sum becomes 19 when the beats are 3, 8, and 8 but it can not be a complete measure because the number of 5 obtained by subtracting 3 from the largest number of 8 is not used.

We developed an algorithm to solve the problem in three steps. As shown in (Figure 4(a)), the first step is to find out if there is a case that it is satisfied by subtracting from the small beat to the size beyond 16. If there is nothing in the smallest number, repeat the



(Figure 3) Beat Postprocessing Problem (a) No Problem with Existing Method (b) Problem with Existing Method



(Figure 4) Three Steps of Beat Postprocessing
 (a) Subtraction from One (b) Subtraction from Two (c) Addition to One and Subtraction from One

same operation with the next number. If all cases are not satisfied, then go to the next step of (Figure 4(b)). At this step, subtracting the number exceeding 16 from two numbers to find out if there are any cases that are satisfied. As shown in (Figure 4 (b)), 3, 8, and 8 have a total of 19, which is 3 larger than 16. So you have to subtract 3, but subtracting 3 from one of 3, 8, or 8 is not possible. In this case, subtracting from two numbers can be used. That is, we make 2 from 3 by subtracting 1 and 6 from 8 by subtracting 2, then 2, 6, and 8 becomes a complete measure. However, as shown in (Figure. 4(c)), subtracting from two numbers sometimes does not solve the problem. In the case of 1, 8, and 8, the total is 17, which must be subtracted by 1, but the two methods above can not be used. In this case, one is added and the others is subtracted. That is, 1 is added to 1 to make 2, and 8 is subtracted from 2 to make 6. Then it eventually becomes 2, 6, 8 and satisfies 16. (There are no cases not to solve the problem with above three steps.) We simulated these three steps for all the possible combinations of numbers, and found that they solved in all cases.

3.2 Rest handling problem

The second problem encountered in previous study is the problem of rest processing. In



(Figure 5) Problem of Rest Processing

order to learn the pitches and rests together in the neural network, the pitches are assigned with numbers from 1 to 36 and the rests are treated as large numbers of 50 so as not to overlap the pitches as described in Section 2. However, problems occurred when composing new songs after learning existing songs. The artificial neural network learned the pitch space from 1 to 36 but also learned the large number of 50 because of the occurrence of the rests. This distorts to the pitch space, and a corresponding distortion occurs in the output pitches during automatic composition. For example, if pitches is progressed to 10, 14, 10, ..., and a rest occurs, the time series data is 10, 14, 10, 50, having a suddenly large value. After learning, if a pitch space is 12, 15, and 9, which is similar to 10, 14, and 10, then the next pitch is mapped to a large number such as 34. In this case, pitches such as 12, 15, 9, and 34 are generated, which hinders the overall melody flow. Thus, the musical pitch space of the melody in the previous method is distorted and composing results are different from the learning pitch space. In the (Figure 5), you can see that the song that normally progresses, but suddenly sounds high as shown in the red circle. To alleviate this problem, we can learn by setting the rest to 37, which is one more than 36, which is the largest number used in notes. In this case, the distortion of the melody pitch space is reduced. However, most of the output in this case is only 1 ~ 36, so there is a problem that

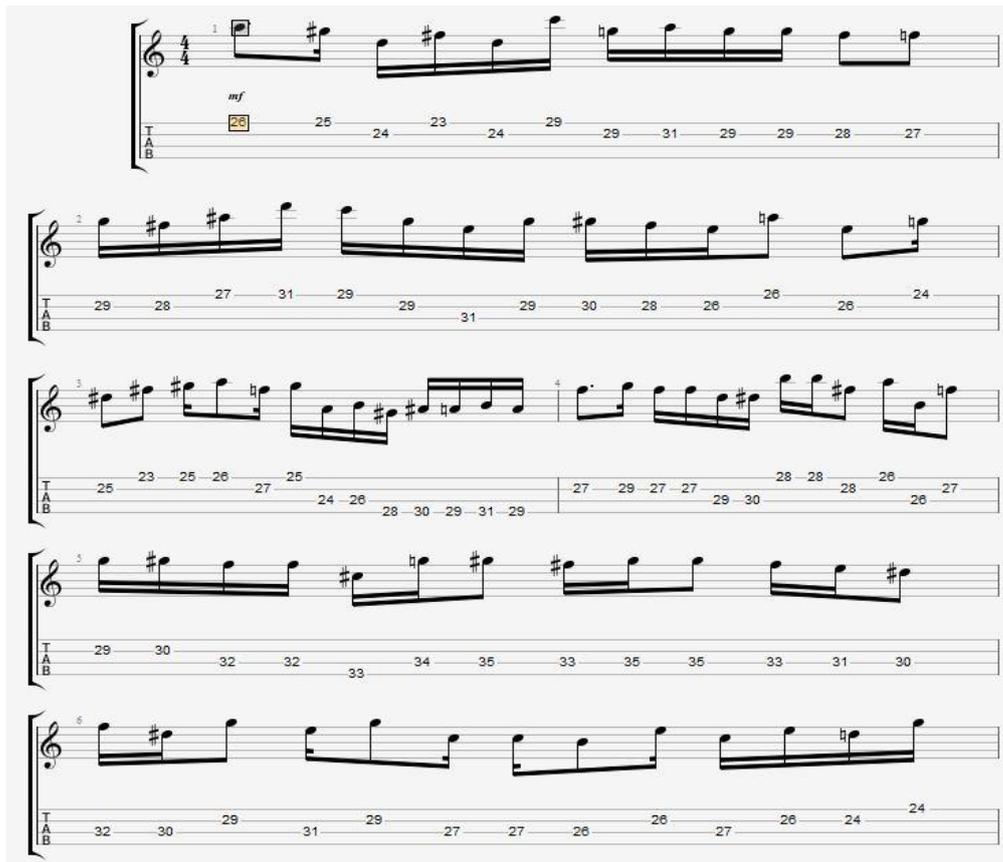
almost no rest is created.

In order to solve this problem, we used two artificial neural networks, one is for pitches and the other is for rests instead of one artificial neural networks for training pitches and rests together. To this end, the rest artificial neural network learns the position of the rest, which is the same as the artificial neural network for training pitches and rests together. Instead, in the neural network for pitches, only the pitches are learned except for the rests. Therefore, a large number 50 for expressing a rest is disappeared in a pitch neural network and there are only the numbers of pitches from 1 to 36. Therefore, only the pitch space can be learned without distorting the pitch space. In artificial neural network for pitches, the rest is assigned by



(Figure 6) Problem of Initial Melody Processing

putting the same value as the previous pitch. If there is a rest at the beginning and so there is no previous pitch, then the rest is changed to the number of the next pitch. As a result of doing this, there was no case that suddenly occurs a high pitch.



(Figure 7) Composition Result of Two Songs with Different Genre

3.3 Initial Melody Processing Problem

The final problem of the previous method is the problem of initial melody processing. If the same initial melody of the learned song is entered, the artificial neural network can make the same remaining melody as the learned melody [1]. However, if different initial melody of the learned song is inputted, it will output a different melody similar to the feeling of the melody. Therefore, in order to compose a new song, a new initial melody must be created and applied by the user [1]. However, since the user can not predict in advance the feel of the melody produced by the artificial neural network, it often occurs that the feeling of the initial melody created by the user is different from the feeling of the remaining melody produced by the artificial neural network. (Figure 6) shows an example. In (Figure 6), the box is the part that the user has composed and the other part is the artificial neural network output. The part of the artificial neural network outputs a song with a certain similar feeling, whereas the part that the user has composed is different from the part of the artificial neural network output. Since the space mapped to the artificial neural network can not be predicted in advance, it is very difficult to make the initial melody similar to that of the artificial neural network.

The initial melody is an indicator of the starting point in the melody space mapped to the artificial neural network. Therefore, the initial melody is used only for starting position of the space to output a new song, and it is not necessary to use the portion for the composition result. So we designed a way to randomly create an initial melody and it is not used in the composition result. As a result, it was possible to solve the problem that the composed music consists of only the output of the artificial neural network, so that the initial melody part and the rest do not match.

4. Experimental results

In order to confirm whether the proposed improvements in this paper have been solved well, We experimented with Yiruma's piano piece "All myself to you" and "River flows in you". (Figure 7) shows the experimental results. As shown in (Figure 7), there is no case that makes incomplete measures where a new beat postprocessing algorithm is applied. And there was no case where the sound was splashed in the middle by using rest artificial neural network. The initial melody was selected randomly and only the output of artificial neural network was used as the composition result except the initial melody. If you do not use the initial melody as a result of composing, you do not have to do it randomly. However, if the initial melody space is very different from the space learned in the artificial neural network, the artificial neural network may switch to the learned space from the initial melody space and affect the initial output to some extent. But the impact is not large, so it will not be critical. You can see in (Figure 7) that the feeling of the song continues from the first. We have done a number of additional experiments in addition to these experiments. In order to test what happen when two different genres of music are trained, we tried to learn "River flows in you" of Yiruma and Roly-Poly of T-ara, which was widely used in previous studies. The two songs were learned by learning the first song and then learning the second song. We tested learning the T-ara's song first and then learning Yiruma's song after that; learning the Yiruma's song first and then learning the T-ara's song after that. (Figure 8) is the result of automatic composition when learning the T-ara's song first and then learning the Yiruma's song. As shown in (Figure 8), the composition result is strongly

The image shows a musical score for guitar. At the top left, it indicates a tempo of $\text{♩} = 120$. The score is written in treble clef for the melody and bass clef for the guitar tablature. The key signature has one sharp (F#), and the time signature is 4/4. The melody starts with a quarter rest, followed by a sequence of eighth and quarter notes. The guitar part is indicated by numbers on a six-line staff, with 'T' for treble and 'B' for bass. The score is divided into four systems, each containing three measures. Measure numbers 1 through 12 are placed above the first note of each measure. The dynamic marking *mf* is placed below the first measure.

(Figure 8) Composition Result of Proposed Method

following the feeling of the secondly trained song. Since the second song is trained after first song, if the second song is strongly trained, then the composition song is strongly following the second song. In order to solve this problem, the second song should be weakly learned, but it is difficult to control the training degree. There are many different ways to do this, including how to mix and match the songs to create learning data and learn how to do it. But, such methods did not produce good results. Future research will need to develop more precise methods of controlling the degree of learning of the two songs, or new methods of learning multiple songs without compromising the quality of the automatic composition results. We performed additional experiments, but in all of the additional experiments, there were no three

problems solved in this paper. Through this, it is confirmed that the proposed method works well.

5. Conclusions

In this paper, we propose a solution for three problems in previous automatic composition research. As a first problem, we propose a new algorithm to solve the problem that the previous beat processing can not be handled in all cases. We confirmed by simulation that the proposed algorithm can be solved in all cases. The second problem is solved by using a rest artificial neural network separately for the problem that the composition song has splashed pitch generated by training rests and pitches in one artificial neural network. Finally, there was a problem

that the initial melody created by the user and the melody generated by the artificial neural network were different from each other. In order to solve this problem, the initial melody was excluded from the result of the final composition and used as a randomly selected value. Many experiments show that all three problems are solved. However, as can be seen from the experimental results, it can be seen that the result of the composition is not composed like a song composed by the composer and is not composed repeatedly. This occurs because the space mapped to the artificial neural network is almost infinite and will need to be solved later.

References

- [1] J. Cho, E. M. Ryu, J. Oh, and S. H. Jung, "Training Method of Artificial Neural Networks for Implementation of Automatic Composition Systems," *KIPS Transactions on Software and Data Engineering*, vol. 3, no. 8, pp. 315-320, Aug. 2014.
- [2] J. Oh, J. Song, K. Kim, and S. H. Jung, "Automatic Composition Using Training Capability of Artificial Neural Networks and Chord Progression," *Journal of Korea Multimedia Society*, vol. 18, no. 11, pp. 1358-1366, Nov. 2015.
- [3] B. Johanson and R. Poli, "GP-Music: An Interactive Genetic Programming System for Music Generation with Automated Fitness Raters" *Proceedings of the Third Annual Conference*, pp. 181-186, 1998.
- [4] N. Tokui and H. Iba, "Music Composition with Interactive Evolutionary Computation," *Proceedings of the Third International Conference on Generative Art*, pp. 215-226, 2000.
- [5] A. Santos, B. Arcay, J. Dorado, J. Romero, and J. Rodriguez, "Evolutionary Computation Systems for Musical Composition," *Proceedings of the International Conference Acoustic and Music: Theory and Applications*, pp. 97-102, 2000.
- [6] C. Chen and R. Miiikkulainen, "Creating Melodies with Evolving Recurrent Neural Networks," *Proceedings of the 2001 International Joint Conference on Neural Networks*, pp. 2241-2246, 2001.
- [7] Debora C. Correa, Alexandre L. M. Levada, Jose H. Saito, and Joao F. Mari, "Neural network based systems for computer-aided musical composition: supervised x unsupervised learning," *Proceeding SAC '08 Proceedings of the 2008 ACM symposium on Applied computing*, pp. 1738-1742, 2008.
- [8] T. Oliwa and M. Wagner, "Composing Music with Neural Networks and Probabilistic Finite-State Machines," *Applications of Evolutionary Computing: EvoWorkshops 2008*, pp. 503-508, 2008.
- [9] H. Kim, B. Kim, and B. Zhang, "Learning music and generation of crossover music using evolutionary hypernetworks," *Proceedings of Korea Computer Congress 2009*, pp. 134-138, 2009.
- [10] G. Bickerman, S. Bosley, P. Swire, and Rober M. Keller, "Learning to Create Jazz Melodies Using Deep Belief Nets," *Proceedings of the International Conference on Computational Creativity*, pp. 228-237, 2010.
- [11] Andr es E. Coca, Roseli A. F. Romero, and Liang Zhao, "Generation of composed musical structures through recurrent neural networks based on chaotic inspiration," *Proceedings of International Joint Conference on Neural Networks*, pp. 3220-3226, 2011.
- [12] J. D. Fernandez and F. Vico, "AI Methods in Algorithmic Composition: A Comprehensive Survey," *Journal of Artificial Intelligence Research*, vol. 48, pp. 513-582, 2013.
- [13] J.-Y. Jo, Y.-H. Kim, M.-K. Lee, "Analysis of Composition Chord Based on Back-propagation Neural Network," *Journal of Digital Contents Society*, vol. 5, no. 3, pp. 245-249, Sep. 2004.



김 경 환

2012년 3월 ~ 현재: 한성대학교 전
자정보공학과 (학부과정)

관심분야: IT, 인공지능



정 성 훈

1988년 2월: 한양대학교 전자공학
과 (공학사)
1991년 2월: KAIST 전기및전자공
학과 (공학석사)
1995년 2월: KAIST 전기및전자공
학과 (공학박사)
1996년 3월 ~ 현재: 한성대학교 전
자정보공학과 교수

관심분야: 인공지능, 시스템생물학, 융합공학