

신경망 기반의 멜로디 작곡법

조재영*, 김윤호*

The Melody Composition by using Neural Network

JaeYoung Jo*, YoonHo Kim*

요 약

본 논문에서는 대중음악 코드진행 과정에 있어서 패턴 분석을 이용하여 멜로디를 추가하는 방법을 소개한다. 먼저, 멜로디를 신경망의 입력으로 사용되는 비트패턴으로 변환하는 방법을 기술한다. 멜로디 추가 방법은 역전파 신경망 학습을 통해 멜로디 작곡 패턴을 학습시키고 학습된 데이터를 바탕으로 멜로디를 생성하도록 설계하였다. 실험결과 신경망 학습을 이용한 컴퓨터의 작곡 가능성을 확인하였다.

ABSTRACT

In this paper, in the middle of progressing popular music chord, a method of inserting melody is addressed, which utilized by analyzing chord progress pattern. Firstly, a method for transforming melody into bit pattern which is to be used for neural network input is described. In order to insert the melody, composition pattern is learned from back propagation neural network, and based on these data new melody is to be generated. Experimental results verified the possibility of neural network based computer composition.

키워드

Chord composition, Melody composition, Chord progress, Neural Network, Back Propagation

1. 서 론

최근의 컴퓨터와 인터넷이라는 매체는 공학과 예술의 분리, 시간과 공간의 제약이라는 기존의 틀을 없애고 있다. 공학과 예술을 접목시키는 시도는 많은 연구 주제로 대두되고 있지만 대부분의 연구는 음원의 내용 분석을 통한 음계정보 추출이라든지 악보인식, 내용기반 검색 등의 기존의 음악이나 악보에 대한 분석적인 목적을 가지는 경우가 많다. 과거 컴퓨터의 자동작곡에 관한 연구는 프랙탈 알고리즘에 의한 작곡이나 코드기반 작곡 등의 시도가 있었고 외국의 경우 카오스 이론을 접목시킨 신경망 학습 방법을 이용한 작곡의 시도가 있었으나 단순 멜로디 학습을 통한 작곡 방법이었기에 대중

적인 작곡에는 근접하지 못한 것이 사실이다.[1]

따라서 여전히 음악 작곡은 작곡가 즉, 인간의 전유물이라고 생각되어지는 경향이 있다. 본 논문에서는 코드진행 패턴분석을 통한 데이터를 바탕으로 코드작곡을 하고 신경망 학습을 이용한 멜로디 작곡을 함으로서 미래의 인공지능 작곡에 관한 가능성을 제안한다.

대중음악은 기본적으로 코드와 멜로디로 구성되어 있다. 멜로디란 음악의 선율을 의미하는 것으로 악보 상에 음표로 기입되어져 있는 음의 높낮이 정보를 의미한다. 반면에 코드정보란 음악의 화음을 의미하는 것으로 마디 내에 구성되어진 선율들에 어울리는 연주를 위해 구성되어진 일반적으로 3개 이상의 음을 동시에 연주하는 것을 의미한다.[2]

* 목원대학교 컴퓨터공학부 박사과정(racemaster@mokwon.ac.kr)

** 목원대학교 컴퓨터공학부 교수(yhkim@mokwon.ac.kr)

음악의 작곡은 크게 멜로디 우선 작곡법과 코드 우선 작곡법으로 나눌 수 있다. 멜로디 우선 작곡법이란 음 하나하나의 나열을 통한 작곡 후에 어울리는 코드를 부여하는 방법이고, 코드 우선 작곡법이란 코드진행을 통한 전체적인 곡의 작곡 후에 그 코드를 벗어나지 않는 범주 내에서 선율을 삽입하는 작곡법이다. 두 가지 작곡법 모두 무엇을 먼저 작곡하느냐의 문제일 뿐 멜로디 정보와 코드정보는 최종 작곡되어진 결과물에는 모두 포함되어 있다.[3]

본 논문에서는 기존의 작곡되어진 여러 곡들의 코드진행 패턴을 분석하고 그 데이터를 바탕으로 최적의 코드진행을 가지는 코드진행을 만들고 진행된 코드진행을 바탕으로 멜로디를 삽입한다.

멜로디 삽입의 방법은 신경망 학습을 통해 기존의 작곡가들의 작곡 성향을 학습시키고 학습된 데이터를 바탕으로 도출된 데이터를 삽입하여 멜로디 작곡을 완성한다.

II. 대중음악 코드진행 패턴 분석

여러 장르의 대중음악은 코드진행을 기반으로 작곡되어 지는데 이런 음악 코드들은 일반적인 진행패턴을 가지고 있다. 코드기반 작곡의 경우 밝은 음색을 표현하는 메이저코드와 어둡고 무거운 음색을 표현하는 마이너코드 등을 적절히 조합하여 작곡되어 진다. 대부분의 사람들이 어려서부터 음악을 들어오면서 성장해왔기 때문에 너무 크게 일반적 에서 벗어나는 음악을 듣는다면 대중들은 어색함을 느끼게 된다. 일례로 C코드 다음에 F코드가 나오는 것은 일반적인 코드진행으로 조화로운 코드진행이지만 C코드 다음에 Bb코드가 나온다면 그것은 일반적으로 사용되지 않는 어색한 코드진행이므로 기존의 음악에 익숙한 대중들은 어색함을 느끼게 된다.[2]

본 실험을 위해서 대중음악 코드를 168개로 분류하고 대중음악 300곡을 분석하여 각각의 코드에서 진행 가능한 다음 코드들을 분석하였다.[3]

다음의 표 1은 대중가요 300곡을 분석하여 얻은 결과 중 C코드 다음으로 진행되었던 코드들을 개수별로 나열한 표이다.[3] 분석결과 C코드 다음으로 11종류의 코드가 나열 되었으며 가장 많이 나열

되는 코드는 G, G7, Am, D7 순서였다. 표 1에서 알 수 있듯이 본 논문을 위해 분석한 300곡의 곡 자료에서 C코드 다음으로 진행 가능한 코드의 개수는 11개 이다. 표 1은 168개의 코드 중 하나인 C 코드 다음으로 나온 코드의 종류와 그 개수를 표현한 표이다. 고려대상이었던 168개의 코드에 대한 모든 진행 가능 코드 종류를 본 내용에서 모두 표현할 수는 없지만 분석결과 실제로 사용되지 않는 코드가 59% 였으며 41%의 코드만 사용 되었다.

표 1. C 코드 다음으로 진행 가능한 코드 종류와 개수

Table 1. Proceed from C chord to other the type and the number of possible chord

C 코드 다음으로 진행되는 코드 종류와 개수						
코드 종류	Am	B7	Cm	D7	Dm	Fm7
개수	44	8	4	32	12	16
코드 종류	Em7	F	F7	G	G7	-
개수	8	20	20	76	60	-

또한 특정 코드에서 진행 가능한 다음 코드의 종류도 15개를 넘는 경우는 없었다. 즉, 대부분의 작곡가들이 사용하는 코드들은 한정적이며 코드진행 또한 어느 정도의 규칙을 가지고 있음을 알 수 있다.

III. 코드작곡 방법

분석된 코드진행 가능경로 데이터를 바탕으로 코드작곡을 완성하기 위해서는 마디 나열 방법에 대한 정의가 필요하다. 그림 1은 대중음악에서의 마디의 표현을 나타내고 있다.

대부분의 대중음악 작곡 결과물의 경우 한 곡은 전주, 간주, 후주를 포함하여 최소 20마디 이상의 길이로 표현된다. 마디 나열 패턴이란 특정 곡 안에서 코드가 나열되는 패턴을 의미한다. 즉, 일반적인 대중음악 작곡 결과물은 전주를 제외한 처음 도입부의 코드진행은 반복되는 경우가 많다는지, 혹은 클라이막스 부분의 코드진행은 간주가 끝난후에 다시 한번 반복된다든지 등의 법칙을 이용해 본

논문에서는 다음과 같은 방법으로 마디 나열 방법을 정의한다.

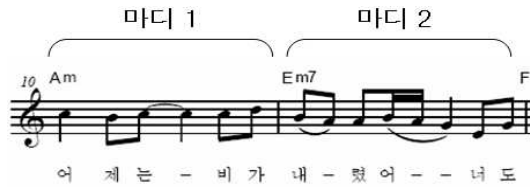


그림 1. 마디의 표현
Fig 1. The expression of a tune

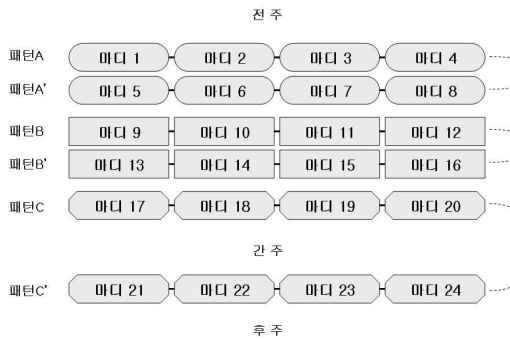


그림 2. 마디나열 패턴
Fig 2. The array tune pattern

전주, 간주, 후주에 해당하는 마디는 각각 4개의 마디를 갖는다. 본 실험에서의 한 곡의 모든 마디 총 수는 36 마디이며 조화로운 코드진행을 나열해 가는 코드작곡 프로그램의 대략적인 알고리즘은 다음과 같다.[5][7][8][9][10]

Step 1. Definition

Melody List : defined chords as array

Syllable: defines syllable as array (total 36 syllables in a song are empty)

Last chord : defines Last chord as variable which gets a previous value(starting chord) for setting following chords

Step 2. Selecting the Start Chord

If the start chord is selected, the program will generate a song contained 36 chords.

Step 3. Processing

If the starting chord is selected and the generate button is pushed, then:

1. Set the chord value from the select menu for the Last chord variable.

2. Set the same value for the first syllable.

3. Find the start chords in the array.

4. If it matched with the selected chord, get a random chord and set the value for the next syllable.

* #3 and #4 are in a while and for loop. All the rest of the transactions below run in the loop.

5. Generate chords for Prelude (the first chord in the Prelude is the chord selected).

6. Generate chords for pattern A (the same as #5).

7. Set A' part same as pattern A.

8. Generate chords for pattern B (the first chord in this part is set from the 2nd chord in pattern A').

9. Set B' part same as pattern B.

10. Generate chords for pattern C (the first chord in this part is set from the 2nd chord in pattern B').

11. Generate chords for Interlude (the first chords in this part is set from the last chord in pattern C).

12. Set C' part same as pattern C.

13. Generate chords for Postlude (the first chords in this part is set from the last chord in pattern C', and the random process will be generating chords until the last chord is matched with the selected chord at the first).

Step 4. Display on the screen

IV. 신경망 기반 멜로디 패턴 학습

서론부에서 설명했듯 모든 대중음악은 멜로디 정보와 코드정보로 구성되어있다. 완벽한 한 곡의 컴퓨터 자동 작곡을 위해서는 조화로운 코드진행을 통한 코드작곡이 끝난 후의 최종 결과물에 멜로디 작곡을 위한 선율정보를 입력해야한다. 본 실험에서는 멜로디 작곡을 위해 신경망 학습 방법을 이용 하였다. 본 연구에서 사용된 Back Propagation Neural Network (이하 역전파 신경망)의 학습은

입력의 작은 값으로 인접층 유니트간의 연결강도를 초기화한 후 학습 데이터들을 입력층에 반복적으로 제공함으로써 학습을 한다. 이 과정에서 입력 벡터와 연결강도에 의하여 계산된 출력층에서의 출력값과 기대하고 있던 목표 출력값과의 차이 즉, 오차를 하위층으로 전파시킴으로써 하위층과의 연결 강도를 재조정한다. 실험에서의 출력층의 연결강도 변화율과 중간층의 연결강도 변화율은 다음 식과 같이 나타낼 수 있다.

$$\delta_k = (T_k - R_k) \cdot R_k \cdot (1 - R_k)$$

.....(1)

$$\delta_j = M_j \cdot (1 - M_j) \sum_k \delta_k \cdot W_{kj}$$

.....(2)

또한, 역전파 학습 알고리즘은 오차가 감소하는 방향으로 연결강도 W_{kj} , W_{ji} 를 재조정하므로 변화량은 다음 식과 같이 나타내어진다.

$$\Delta W_{kj} \cong \frac{-\partial E}{\partial W_{kj}} = \eta \cdot \delta_k \cdot y_j$$

.....(3)

$$\Delta W_{ji} \cong \frac{-\partial E}{\partial W_{ji}} = \eta \cdot \delta_j \cdot V_i$$

.....(4)

여기서 η 는 학습계수(Learning constant)이며 신경회로망의 학습 과정을 서서히 진동 없이 수렴시키기 위해서 모우멘텀(Momentum) α 를 넣으면 $n+1$ 단계에서 연결강도의 변화량은 다음 식과 같이 정리된다.

$$\Delta W_{kj}(n+1) = \eta \cdot \delta_k \cdot y_j + \alpha \cdot \Delta W_{kj}(n)$$

.....(5)

$$\Delta W_{ji}(n+1) = \eta \cdot \delta_j \cdot V_i + \alpha \cdot \Delta W_{ji}(n)$$

.....(6)

이와 같이 역전파 학습 과정을 거쳐 각 층간의 연결강도를 계속하여 재조정해 나가면서 출력층의 오차가 허용범위 이내의 값으로 수렴하게 되면 학

습이 완료된다.

- k : 출력 유니트
- j : 중간 유니트
- T_k : 출력 유니트 k의 목표 출력값
- R_k : 출력 유니트 k의 실제 출력값
- M_j : 중간 유니트 j의 출력
- δ_k : 출력 유니트 k의 연결강도 변화율
- δ_j : 중간 유니트 j의 연결강도 변화율

본 논문에서의 신경망 학습 방법 실험을 위해 기존의 기보법(악보를 표현하는 방법)을 오선 기보법이 아닌 다른 방식으로 표현 할 필요가 있었으므로 비트맵 행렬로 표현하는 방법을 설명한다. 일반적으로 가장 많이 쓰이는 오선 기보법을 비트맵 행렬 기보법으로 표현하는 방법을 다음과 같이 진행한다.

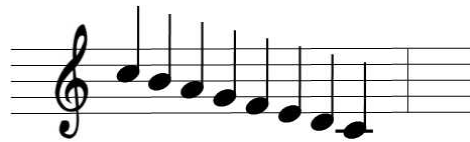


그림 3. 오선 기보법의 “도시라솔파미레도” 표현

Fig 3. The expression for musical scale “Do to Do” through the stave music notation.

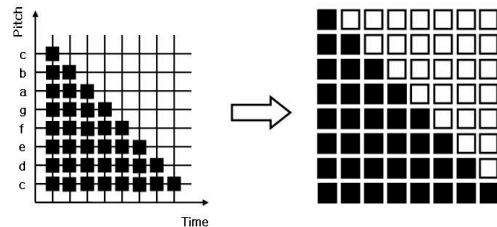


그림 4. “도시라솔파미레도”의 비트맵 행렬 표기방법

Fig 4. The bitmap procession marking way of the musical scale “Do to Do”.

그림 4는 그림 3의 “도시라솔파미레도”를 비트맵 행렬 기보법으로 표기 한 그림이다. X축 좌표는 음표의 길이를 나타내며 Y축 좌표는 음의 높낮이를 나타낸다. 본 실험에서는 반음에 해당하는 음은 제외하였으며 멜로디 음역은 2 옥타브로 한정한다. 음표의 길이도 4분음표와 8분음표로 한정시켰

다.[4][6]

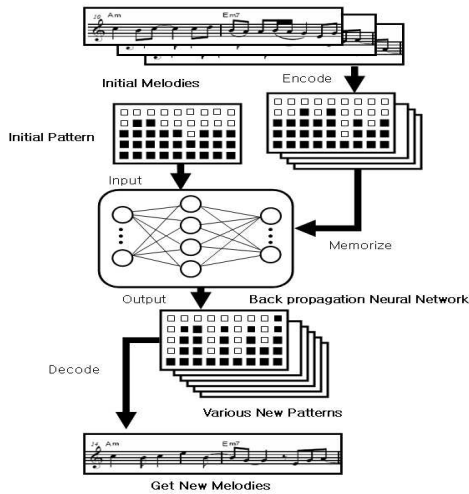


그림 5. 역전파 신경망을 이용한 멜로디 패턴 생성 알고리즘 블록도

Fig 5. The block diagram of melody pattern generation using backpropagation neural network

그림 5는 신경망 학습을 통한 멜로디 패턴 생성에 대한 전반적인 진행에 대한 그림이다. Am 다음 코드로 Em7이 진행되었을 경우를 예로 든 그림으로 해당 코드진행을 가지는 멜로디 패턴을 작곡가 박진영의 곡들 중에서 검색하여 해당 작곡 패턴을 비트맵 행렬로 인코딩 후 신경망 학습시킨다. 신경망 학습의 방법으로는 역전파 알고리즘을 사용하였으며 최종적인 신경망 학습이 끝난 후 학습전의 input layer에 주어졌던 멜로디와 다른 패턴을 부여했을 경우 본 시스템은 학습된 데이터를 바탕으로 새로운 패턴의 멜로디를 생성한다. 본 시스템에서는 랜덤코드를 이용한 코드 작곡 후에 코드 작곡된 결과물을 바탕으로 신경망 학습을 통한 멜로디 작곡을 한다. 먼저 학습을 위한 복수의 멜로디 패턴을 준비하고 실험을 위해 멜로디 패턴은 0과 1로 코딩되어 신경망 학습을 위한 고정 패턴으로서 이용된다. 본 실험에서는 작곡가 “박진영”의 곡들을 대상으로 멜로디 패턴을 채취하였으며 신경망 학습을 이용한 멜로디 작곡 방법은 다음과 같은 순서로 이루어진다.

1. 조화로운 코드 나열 프로그램을 통한 코드작

곡을 완성하고 코드진행 패턴을 분석하여 멜로디 학습 대상이 될 멜로디 패턴을 해당 작곡가의 곡에서 검색한다.

2. 학습시킬 작곡가 박진영의 멜로디 진행 패턴을 분석하기 위해 고려 대상이 되는 마디의 멜로디를 비트맵 행렬로 인코딩 한다.

3. 신경망 학습을 위해 채취한 멜로디 표본을 바탕으로 특정 코드에서 진행되는 다음 코드로의 상황을 신경망 학습시킨다.

4. 학습된 데이터를 바탕으로 첫 멜로디로 다른 멜로디가 주어졌을 경우 새로운 다음 코드의 멜로디를 생성한다.

5. 생성된 멜로디 데이터를 조화로운 코드진행에 맞게 나열하여 멜로디 작곡을 완성한다.

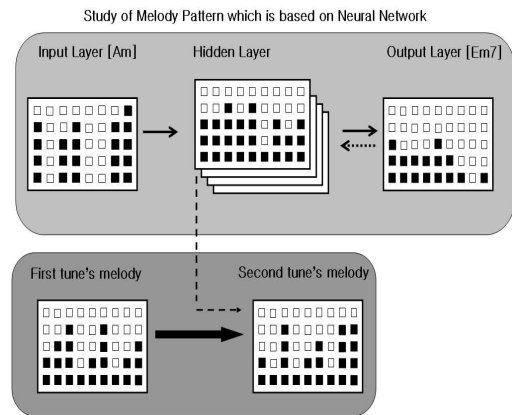


그림 6. 역전파 신경망 학습을 통한 멜로디 생성
Fig 6. Melody pattern generation using backpropagation neural network

그림 6는 Am 다음 코드로 Em7 코드가 나오는 상황에 대한 학습을 표현한 그림이며 input layer는 그림 1의 첫 번째 마디의 멜로디를 비트맵 행렬로 입력 시킨 것을 의미하며 output layer는 그림 1의 두 번째 마디의 멜로디에 해당한다. 즉, 작곡가 박진영의 작곡 패턴을 학습시키고 첫 번째 마디의 멜로디로 초기 input 값과 다른 값을 부여했을 경

우 학습된 데이터를 바탕으로 새로운 패턴의 멜로디를 생성하게 된다. 그림 7은 코드생성 프로그램을 통해 코드작곡이 끝난 코드작곡 결과물에 신경망 학습을 통한 멜로디 생성까지의 실험을 끝낸 최종 결과물 중 첫 4마디의 출력 결과를 보여주는 그림이다. 코드작곡 프로그램을 통해 Am-Em7-A7-Dm 라는 코드진행이 생성된 후 해당 코드진행들의 멜로디들을 특정 작곡가의 곡들 중에서 추출하여 신경망 학습을 거치면 첫 Am 코드에 새로운 멜로디를 부여 할 경우 학습된 데이터를 바탕으로 자동작곡을 시행하게 된다.

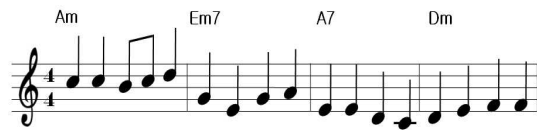


그림 7. 멜로디 학습을 통한 새로운 멜로디 생성
Fig 7. Creating new melody based on the learning melody

V. 결과 및 향후 방향

본 논문에서는 기존의 작곡되어진 여러 대중음악의 코드진행 패턴을 분석하고 분석된 데이터를 바탕으로 최적의 코드진행뿐만 아니라 다양한 코드진행의 코드 작곡을 하였다. 또한 신경망 학습을 통해 특정 작곡가의 멜로디 작곡 성향을 학습시켜 새로운 멜로디 패턴을 작곡 하였다. 그 결과 특정 작곡가의 작곡 성향을 가진 새로운 멜로디 패턴을 생성 할 수 있었다. 향후 방향으로는 반음까지 고려 가능한 시스템의 구축과 4분음표와 8분음표 뿐만 아니라 그 외의 모든 음표를 대상으로 학습 가능한 방식의 시스템의 구축을 목표로 한다.

참 고 문 헌

[1] 윤중선, “알고리즘에 의한 음악의 작곡”, Proceedings for the KACC, 1997.
[2] 길옥윤, “경음악 편곡법”, 1988.
[3] 박세원, “대중가요 대백과”, 2002.
[4] 도용태 김일곤 김종환 박장현, “인공지능 개념 및 응용”, 사이텍미디어, 2001.

[5] Chen chunling, Wang shaodi and shan bingzhe, “A Fractal Image Coding Based on The Quadtree.” Proceedings of ICSP , 1998.
[6] TOKUMARU, Masataka, MURANAKA, Noriaki, IMANISHI, Shigeru, “Memory and Forgetfulness : Application to A Music Composition System Using Chaotic Neural Network with Memorizing Patterns”, Journal of Japan Society for Fuzzy Theory and Intelligent Informatics, 2007, Vol.v.19 no.3, No., Page.299~312
[7] 1604-VLZ PRO 16-channel owner’s Manual, 2000.
[8] PC2 Kurzweil Musician’s Guide, 2002.
[9] David Cope, “Virtual Music”, 2001.
[10] 조재영, 김윤호, “코드기반 작곡을 위한 트리구조 프로그램 구현”, 해양정보통신학회, 2003.

저자약력



조 재 영(Jae-young Jo)

2002년 3월 목원대학교 컴퓨터공학부(공학사)
2004년 3월 목원대학교 컴퓨터공학부(공학석사)
2008년 12월 목원대학교 컴퓨터 공학부 박사과정

<관심분야> 디지털 오디오 시스템, Engineering Art, 영상편집, 게임제작



김 윤 호(Yoon-ho Kim)

1992~현재 목원대학교 컴퓨터공학부 교수

<관심분야> 영상처리, 컴퓨터비전, 뉴로퍼지응용, 정보통신정책 등.