Generation of composed musical structures through recurrent neural networks based on chaotic inspiration

Andrés E. Coca, Roseli A. F. Romero, and Liang Zhao

Abstract—In this work, an Elman recurrent neural network is used for automatic musical structure composition based on the style of a music previously learned during the training phase. Furthermore, a small fragment of a chaotic melody is added to the input layer of the neural network as an inspiration source to attain a greater variability of melodies. The neural network is trained by using the BPTT (back propagation through time) algorithm. Some melody measures are also presented for characterizing the melodies provided by the neural network and for analyzing the effect obtained by the insertion of chaotic inspiration in relation to the original melody characteristics. Specifically, a similarity melodic measure is considered for contrasting the variability obtained between the learned melody and each one of the composite melodies by using different quantities of inspiration musical notes.

I. INTRODUCTION

Recurrent neural networks (RNN) considers feedbacks and delay operators, which allows to model the nonlinearity and dynamical components of a system. They are very useful in the analysis and modeling of time series, such as music. Therefore, we can find many works that applies recurrent neural networks for automatical music composition. In 1989, Todd [1] utilized a neural network trained by BPTT algorithm, for composing monophonic melodies, in which a note occurs after other as a sequential phenomenal. In [2], it was developed a paradigm that utilizes neural network for the generation of the melodies by refinement (CBR - creation by refinement), in which the network was trained for performing a musical critic to judge musical examples according to specified criterion.

Another interesting system for musical composition is the CONCERT system, proposed by Mozer [3]. In that work, a recurrent neural network was trained based on the psychological information, that is, similar notes are represented by similar representation in the network. The system presents a peculiar characteristic in which the generated notes are candidates to belong to the melody according to a probability distribution.

It is observed that in the past years, the methods for musical composition were evolving and turning themselves a mixture of computational methods, as it can be found in [4]. In that work, it was used a recurrent neural network, SRN - simple recurrent networks, in which the weights are adapted according to a genetic algorithm.

Recently, hybrid methods have been proposed as in [5], in which neural networks and Markov Chain are used for musical composition through examples. In [6], it was used also a neural network with delay time and probabilistic finite state machines for acquiring knowledge by inductive learning for multiple musical instruments. In [7], a recurrent neural network was proposed to model the multiple threads of a song that are functionally related, such as the functional relationship between the instruments and drums. Finally, in [8], a probabilistic neural network based on Boltzmann machines was proposed for melodic improvisation of jazz on the sequential accords.

It is possible to obtain a similar melody from an original training melody by using a neural network to learn the characteristics or style of a input melody. It is also interesting to control the similarity between the original melody and the generated melody (or output melody of the neural network). In this case, it is necessary to consider an additional independent melody. In [10], an inspiration source based on the texture of the geographic landscapes is used. However, this method has some disadvantages. The first one is the necessity of the creation and storage of images of a data set, that occupy a large quantity of memory; the second one is about the necessity of a preprocessing of each image, which is time consuming.

One of the main characteristics of the chaotic dynamic systems is the high sensitivity to the initial conditions, which is a capacity of a dynamical system to evolve in a different or even a totally unexpected way due to small changes in the initial condition of the system. During the last few decades, this property woke up great interest in different areas of the research, including the musical composition.

In this work, we propose to use melodies generated by a chaotic dynamical system as inspiration source. By using this technique, it is not necessary to have a large dataset for getting a big number of inspiration melodies. Due to the characteristics of chaotic systems, it is possible to get infinite variations through a arbitrarily small changing in the parameters of the system. Furthermore, we can get rich musical materials in a very fast manner. For this purpose, we use the algorithm proposed by Coca et. al. [11] for chaotic musical composition as it will be shown later.

The rest of this paper is organized as it follows. In section II, the theoretical foundations of Elman neural networks and the chaotic inspiration are presented. In section III, we describe the algorithm for generation of chaotic musical inspiration. In section IV, we present the melodic measures used in this work in order to analyze and to compare the
melodic similarity and complexity. Some simulation results obtained are showed in section V. Finally, section VI presents the conclusions and future works.

II. RECURRENT NEURAL NETWORK WITH CHAOTIC INSPIRATION

Elman Networks are recurrent neural networks where the feedback occurs from the output of each neuron of the hidden layer for all the neurons of the same layer. Another layer, called the context layer, simulates the memory of the network. Figure 1 shows the modified Elman network which contains an additional input, which is generated by an algorithm of chaotic musical composition.

![Fig. 1. Recurrent Neural network with chaotic inspiration](image)

The processing of the network consists of the following events: at iteration 0 (initial), the signal is propagated by the network. After that, the context units initiate with the output of the hidden layer with value 0, in order to not influence the output of the network. That is, in the first iteration, the network will behave like a feed-forward network [12]. At iteration \( t \), the hidden neurons will activate the neurons in the context layer and these will store the output of this iteration that will be used in the next cycle. Then the backpropagation algorithm is applied for the correction of the synaptic weights \( \mathbf{W} \), with exception to the recurrent synapses that are fixed in the value 1. At the next iteration, \( t + 1 \), this process is repeated. from now on, there is a difference: the hidden neurons will be activated by both the input units and the context layer units, which have the value of the hidden neuron outputs at instant \( t \). The output the neural network is given by Equation 1

\[
\begin{align*}
\mathbf{x}(t) &= \mathbf{W}_{xx} \mathbf{x}(t - 1) + \mathbf{W}_{xu} \mathbf{u}(t - 1) + \mathbf{W}_{zu} \mathbf{z}(t - 1) \\
\mathbf{y}(t) &= \mathbf{W}_{yx} \mathbf{x}(t)
\end{align*}
\]  

(1)

where, \( \mathbf{x}(t - 1) \) are the outputs of the hidden neurons, \( \mathbf{W}_{xx} \) are the associated synaptic weights, \( \mathbf{y}(t) \) are the network outputs, \( \mathbf{u}(t - 1) \) are the network inputs and \( \mathbf{z}(t - 1) \) are the inputs of chaotic inspiration generated for the algorithm of composition described in section III.

The representation by cycles of thirds is used to represent the input data in the form of bits [9]. This representation uses seven bits, in which the first four bits indicate in what cycle of major third the note is located, for a total of four cycles. The last three bits indicate in what cycle of minor third the note is located, for a total of three cycles. The information of octave is given separately, with two additional neurons, one to indicate if the octave is of C1 until B3, other to indicate if the octave is of C5 until B5. If the two neurons have values equal to zero, then the octave is of C3 until B3. Figure 2 shows the four cycles of major third used and the three circles of minor thirds. Each cycle is read in the anti-clock direction.

![Fig. 2. Major and minor thirds circles [9].](image)

III. CHAOTIC INSPIRATION ALGORITHM

The algorithm developed in [11] uses the numerical solution of a nonlinear dynamical system, which consists of three variables \( x(t), y(t), \) and \( z(t) \). The first variable \( x(t) \) is associated with the extraction of musical pitches for the input inspiration. Data transformations of this variable are described in the following subsections.

A. Musical Scale Specifications

Next, the initial inputs to the chaotic musical composition algorithm are described.

1) **Number of Octaves** \( k \): indicates the extent of the scale in octaves, where \( k \in \mathbb{N} \) and \( 0 < k \leq 7 \).

2) **Tonic** \( T_{\tau,o} \): is the initial tone where a scale starts.

   It is defined as a pair of variables \( (\tau,o) \), where \( \tau : \{\tau \in \mathbb{N} | 1 \leq \tau \leq 12\} \) is the tone and \( o : \{o \in \mathbb{N} | o \leq k\} \) is the number of the octave of the user defined scale.

3) **Mode** \( m_0 \): is a value within the range \( 0 < m_0 \leq m \), where \( m \in [0,11] \) is the maximum number of possible modes for a given scale and \( m_0 \) indicates the number of required shifts in which a scale starts with a tone given by \( T_{\tau,o} \).

4) **Structure of the Scale** \( \psi_\xi \): a set of interval generators that form the architecture of the musical scale \( \xi \). It is represented by the set \( \psi_\xi = (s,t,t_m) \), where \( s \in [0,12] \) is the number of semitones, \( t \in [0,6] \) is the number of tones, and \( t_m \in [0,4] \) is the number of one-half tones that constitute the structure of the musical scale \( \xi \) with \( n \) notes.
5) **Tone Division** $\Delta$: represents the number of divisions by which the diatonic tone of the scale is divided. For instance, 2 divisions ($\Delta = 2$) are used by the tempered system of 12 notes, 3 divisions are used by the third tone scale, 4 divisions are used by the quarter tone scale, and so on.

**B. Extraction of Frequencies and Musical Notes**

The extraction of frequencies and musical notes is divided into three steps. In the first step, it is generated a membership binary vector indicating whether each musical note belongs to the scale $\xi$. In the second step, a matrix $D$ is constructed by eliminating all zero entries of the vector $r$, keeping only those intervals belonging to the user defined scale $\xi$ and eliminating the others. The result of this operation is the vector $e$,

$$e_i = v_i \cdot r_i, \quad 0 \leq i \leq p + 1$$

(4)

Finally, the vector $g$ with dimension $n$ is obtained ($n \leq p + 1$ is the number of musical notes of the scale $\xi$), which is constructed by eliminating all zero entries of the vector $e$, defined as:

$$g = \{ e_i | e_i \neq 0, \forall i \}$$

(5)

It can be seen that $g$ contains only the frequency ratios of the specific musical scale $\xi$.

**Step 2: Variable Normalization**

The variable $x(t)$ is selected to generate the frequencies of the musical notes. All values of $x(t)$ in the chosen interval should be normalized with respect to the vector $g$, because the range of the variable $x(t)$ is different from the range of the vector $g$. This normalization process, defined as $x_n(t) = \gamma(x(t), g)$, consists of a scaling and a translation of the numerical solution of the variable $x(t)$ adapted to the vector $g$, obtaining the normalized variable $x_n(t)$.

This normalization process is responsible for the adjustment of the maximum and minimum values between the two groups of data, i.e., $\max(x_n(t)) = \max(g)$ and $\min(x_n(t)) = \min(g)$, while keeping the proportion of intermediate data. The normalization is defined as:

$$x_n(t) = \alpha x(t) + \beta$$

(6)

where $\alpha$ is a scaling factor, calculated by the following formula (note that $\max(g) = 2^k$ and $\min(g) = 1)$:

$$\alpha = \frac{2^k - 1}{\max(x(t)) - \min(x(t))}$$

(7)

Similarly, the variable $\beta$ is a translation factor and it is determined by the following equation:

$$\beta = -\alpha \min(x(t)) + \min(g) = -\alpha \min(x(t)) + 1$$

(8)

In this way, we get a variable in the range $1 \leq x_n(t) \leq 2^k$.

**Step 3: Mapping to the Closest Value**

Once the normalized variable $x_n(t)$ is obtained, for each value $x_n(t)$, the closest value in the vector $g$, getting a match between $x_n(t)$ and the notes of the given musical scale $\xi$. Next, a matrix $D$ of dimension $c_{x_n} \times n$ is built, such that $c_{x_n}$ represents the number of elements of a piece of numerical solution of $x_n(t)$ and $n$ is the number of musical notes. This matrix is constructed according to Eq. (9) with the indices $i$ and $j$ in the ranges $0 \leq i \leq c_{x_n}$ and $0 \leq j \leq n$, respectively.

$$D_{i,j} = \begin{cases} 0, & \text{if } |x_n_j(t) - g_i| \leq \mu \\ x_n_j(t), & \text{if } |x_n_j(t) - g_i| > \mu \end{cases}$$

(9)

The threshold value $\mu$ depends on the type of tuning factor $\lambda$, which is calculated by

$$\mu = 2^{\lambda} - 1$$

(10)

Then we generate a new vector $h$ of size $c_{x_n}$, which holds the position of the minimum value of each row of the matrix $D$, so

$$h_i = col(\min D_i), \quad 0 \leq i \leq c_{x_n}$$

(11)

where $D_i$ is the $i$th row of matrix $D$ and $col(\min D_i)$ returns the column index of the minimum value of $i$th row of $D$. In other words, the vector $h$ contains the column indexes of $D$ rather than the values of $D$.

The knowledge of tonic is required for the conversion of the variable $x_n(t)$ to the musical space. The tonic frequency $\delta_{\tau,o}$ with the musical tone $\tau$ in the octave $o$ can be obtained as follows:

$$\delta_{\tau,o} = 55 \cdot 2^{\frac{\tau - 120 - 10}{12}}$$

(12)

With the indexes of $h$ and frequency of tonic $\delta_{\tau,o}$, we can calculate the frequencies of musical notes corresponding to the variable $x_n(t)$ by using

$$f_i = \delta_{\tau,o} \cdot h_i, \quad 0 \leq i \leq c_{x_n}$$

(13)
In order to view the score of the melody, these frequencies must be converted to the values of musical notes of the standard MIDI (Musical Instrument Digital Interface). For this purpose, we use Eq. (14), where $f_i$ is a frequency in Hz and $x_i$ is a MIDI value.

$$x_i = 69 + \frac{12}{\log 2} \log \left( \frac{|F_i|}{440} \right)$$

Making the conversion to standard MIDI by using the values of frequencies $f_i$, a vector $x$ of dimension $c \times n$ is obtained, which contains the numbers of pitches generated by $x(t)$. The whole melody generating process by using a chaotic system is summarized in Fig. 3. It should be noted that only part of the diagram is used in this paper, which includes all the steps on the left side of the diagram, the step “Generate Matrix of Notes” and the step “Write to the MIDI file”.

IV. MELODIC MEASURES

In this section, the measures to quantify the characteristics of the composed melodies for the network are described. It is useful to find the relation between the musical characteristics of the melody generated by the network and the quantity of inspiration notes.

A. Expectancy-based model of melodic complexity

This model is based on the melodic complexity expectancy because the components of the model are derived from the melodic expectancy theory [14]. This measure has as reference the collection of Essen which has a complexity mean of 5 with a standard deviation of 1 [15]. The melodic complexity can be tonal (CBMP), rhythmic (CBMR) or joint (pitch and rhythm, CBMO). These values have been found for the corresponding scales of predictability given by listeners in experiments [16].

B. Melodic originality

In the studies performed by Dean Keith Simonton between 1984 and 1994, after analyzing a great amount of classic themes, he concluded that the originality of the themes is connected to their complexity. The relationship between originality and complexity has an inverted U shape, i.e., the simplest and the most complex melodies are less originality and melodies with average complexity have the highest originality. In addition, the originality is directly proportional to the popularity. In other words, the themes most popular have medium originality. Consequently, the most simple and the most complex themes are not considered popular. Accordingly, the originality is lower.

The output of the model is the inverse of the averaged probability, scaled in the interval from 0 to 10, where higher value indicates higher melodic originality (MOM) [16].

C. Melodic similarity

This measure is indicated to quantify the similarity between melodic motives, phrases and musical segments [17]. The distance can be calculated from a melodic representation as a distribution and a proximity measure, for example the distribution of pitch class (set of notes used in the melody) and the taxi cab distance [16].

The similarity measure is scaled in the range from 0 to 1, where 1 indicates the highest similarity. This measure can be determined in relation to a property of the melody as pitch, rhythm or dynamics. Reason for such, a melody can have high rhythmical similarity, but low similarity of the pitch with other melodies [14]. The measures of similarity used in this work and their meanings are described as follows:

1) Melodic similarity according to the distribution of pitch class: it measures the similarity between the notes that conform the set of pitch class and the distribution of the number of times that each note appears in the set.

2) Melodic similarity according to the distribution of durations: it measures the similarity of the frequency of appearance of the durations that compose the musical phrase.

V. EXPERIMENTAL RESULTS

In this section, the experiments and the obtained results are described. The objective is to analyze the effect that has the inclusion of chaotic inspiration notes into the training phase to generate melodies. A recurrent neural network constituted
by a hidden layer of 20 neurons are used. The BPTT (back propagation through time) algorithm with momentum term is applied to train the network. The learning rate is 0.1 and the momentum parameter is 0.01. The network is trained using four notes of the original melody. In the simulation phase, we also use four input notes, then the network must compose the next four notes. For the chaotic inspiration, we apply the logistic map in the chaotic regime ($r = 3.9$), varying the number of the notes of inspiration from 0 to 20 in the training phase.

Figure 4 shows the first compasses of the second movement of Vivaldi’s Four Seasons. This melody is used as input to train the recurrent neural network.

![Fig. 4. Compasses of the second movement of the Vivaldi's Four Seasons](image)

The melody shown by Fig. 5 is composed by using algorithm BPTT without notes of chaotic inspiration and using the representation cycles of thirds.

![Fig. 5. Melody composed by algorithm BPTT without notes of chaotic inspiration](image)

The melody shown in Fig. 6 is composed by using algorithm BPTT and four notes of chaotic inspiration.

![Fig. 6. Melody composed by algorithm BPTT with 4 notes of chaotic inspiration](image)

In Table I, one can see the obtained measures of Pitch Complexity (cbmp) and melodic originality (mom) for the training melody (Melody T.), for the composed melody without chaotic inspiration (Melody (0)), and for the melody using one note of chaotic inspiration (Melody(1)). The means and standard deviations are also shown in the same table.

In Table II, one can see the melodic similarity measure between the training melody and the melody composed with Melody(0) and the melody using one note of chaotic inspiration Melody(1), respectively. The means and standard deviations are also shown in the same table.

Figure 7 shows the melodic complexity of the melody composed with different amounts of chaotic inspiration notes.

The dotted line is the complexity of the melody without inspiration notes and the solid line is the complexity of the original melody. It can be noted that complexity of the melody composed by the network is increased proportional to the number of inspiration notes, and it is going to be closer to the value of the complexity of the original melody. Then the melodic complexity of the output melody is proportional to the number of the inspiration notes. Nevertheless, the number of inspiration note increases with the training time, which is an important point to be considered at the time of choosing the proper number of notes to train the network.

For the case of melodic originality, we have found that the originality of the melody composed by the network with chaotic inspiration notes lies close to the value of originality of the training melody, which means that the number of notes of chaotic inspiration does not affect the originality of the output melody, as it is shown in Fig. 8.

Varying the number of the chaotic inspiration notes, it is possible to see that the melodic complexity gets higher when the number of notes inspiration is high and additionally the composed melody is more different with regarding to the training melody. That is, the melody composed by the network becomes less resemblance to the original one if more
notes of chaotic inspiration are used. In Fig. 9, the graphical
of melodic similarity according to the distribution of pitch
between the original melody and each one of the melodies
composed by the network is presented. In this figure, the
number of notes of chaotic inspiration is varied from 0 to 20.
The dotted line is the similarity measure between the original
melody and the melody without chaotic inspiration. The solid
line is the evolution of the melodic similarity between the
training melody and each one of the composed melodies
using different number of chaotic inspiration notes.

Figure 10 shows that the rhythmic similarity of the melody
without chaotic inspiration is high. Moreover, the distance of
similarities between the melody without chaotic inspiration
and with chaotic inspiration gets larger as the number of
inspiration notes is increased.

VI. CONCLUSIONS

In this paper, a recurrent neural network trained by BPTT
is used for generating melodies. It has been added into the
input layer a small fragment of a chaotic melody, considered
as an inspiration source to attain a big variability of melodies.
Several measures are used to quantify the characteristics
of the composed melodies by the neural network. Various
testing are conducted in order to find the relations between
the musical characteristics of the generated melody by the
neural network and the quantity of inspiration notes. The
first of them is about the inclusion of chaotic inspiration
notes in the training phase. It is noted that it has the effect
to change the melodic complexity of the melody composed
by the neural network in the application phase. This is
because the network learns the characteristics of the training
melody with the chaotic inspiration. More notes of chaotic
inspiration we have, the influence of chaos gets stronger,
then one obtains a melody with higher variations, while
preserving the characteristics of the original melody. Thus,
for increasing the differences between the training melody
and the composed melody, it is enough to have a distribution
of notes which is more complex, but less similar to the
original melody. The second one is related to the using of

A. Melodic Originality and Learning Rate

Figure 11 shows the melodic originality by varying the
learning rate. We see that the melodic originality has few
changes for different values of the learning rate.
neural networks, which is possible to learn the characteristics and the inheriting patterns of the training melody, besides of presenting the statistics information adequately of the training set. It is shown that variations in the rate of learning do not affect significantly the melodic originality of the composed melody. As future works, we intend to test the network behavior with different types of chaotic dynamical systems, continuous and discrete, and with different number of variables and parameters, aiming to determine which system is most suitable to compose melodies more “creative” without losing the essence of the main melody of training. In addition, we would like to determine the number of notes of inspiration for adjusting the complexity of the melody of the network within a high range of variability, and then to let the number of notes of inspiration fixed to control the complexity by varying a parameter of the system.

REFERENCES