Finding Tone Patterns in Chinese Poems Using a Simple Recurrent Neural Network

Shuang Liu

The Ohio State University

Introduction

Because of the unique structure of a feedback loop, recurrent neural networks are said to have memory and are able to find out information in a time sequence. Tone sequences in Chinese poems were simulated and learned by a simple recurrent network (SRN). Tone sequences were simulated to follow certain underlying time patterns and rules of tone grouping. Results indicate that an SRN is able to give clues about the underlying patterns and tone grouping after training.

Background

Chinese is a tonal language that has rich tone inventories, especially in some of its dialects. Mandarin is one of its most widely spoken dialect. Mandarin has four tones while Cantonese, another widely spoken dialect, is considered to have six tones. Lexical tones perform a crucial role in differentiate meanings for a tonal language. In Chinese, one syllable can be pronounced with multiple tones to gain multiple meanings.

Each tone is characterized by a special fundamental frequency, or pitch, track. For example, one tone of Mandarin has a flat pitch track, while another tone's pitch rises over time. Typical pitch tracks of tones in Mandarin are depicted in Figure 1.

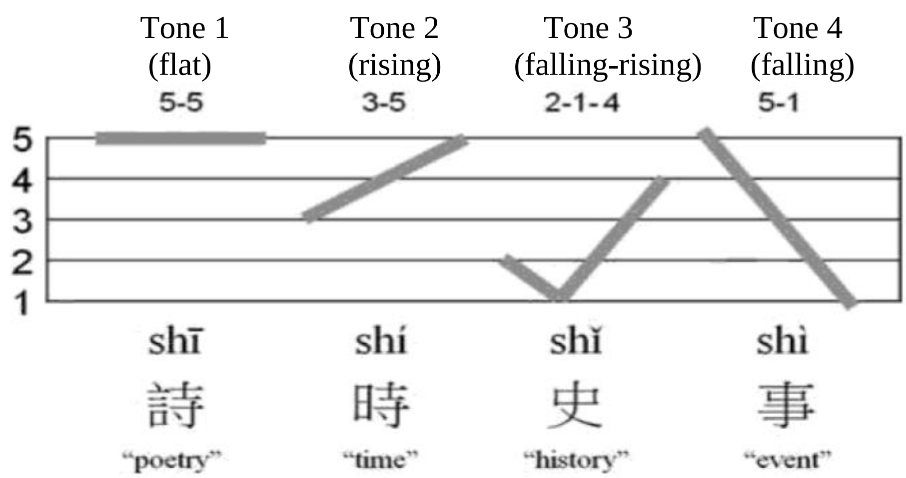

Figure 1. Simplified pitch tracks of tones in Mandarin. Numbers indicate relative pitch. Reprinted from *Qupai in Chinese Music: Melodic Models in Form and Practice.* (p. 143), by A. R. Thrasher, 2016, Routledge.

Tones contribute to the musical property of tonal languages. Researchers claim that to make a Chinese poem sound pleasant or be easy to recite, the tone sequence needs to follow certain underlying patterns (Thrasher, 2016). These patterns are commonly called "Ping-Ze" pattern. In this paper, I will not use the term "Ping-Ze" pattern, since it sounds too complicated and this complication is not necessary. Instead, I will use "green-purple" pattern to refer to "Ping-Ze" pattern. Figure 2 shows an example.

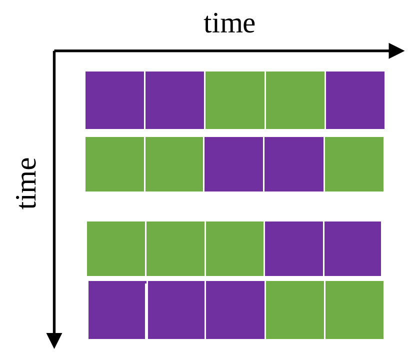

Figure 2. An example of green-purple pattern (Ping-Ze pattern).

The basis of this pattern is the classification of tones into two categories: Green (ping) and Purple (Ze). In Mandarin, tone 1 and tone 2 are categorized as green tones (Ping tones: pitch is relatively stable). Tone 3 and tone 4 are categorized as purple tones (Ze tones: pitch is unstable). A green block in the pattern above means that only tone 1 or tone 2 can appear at this position, while A purple block means only tone 3 or tone 4 can appear at this position. An example of a poem that follows this pattern is shown in Table 1.

2 4 1 1 4	白日依山尽
2 2 4 3 2	黄河入海流
4 2 1 3 4	欲穷千里目
4 4 4 2 2	更上一层楼

Table 1. A poem that follows the green-purple pattern. Each number on the left indicates the tone of each character/syllable on the right.

It is very clear that most tones in this poem follow the pattern very well. The only two exceptions are the first character of the first line and the first character of the third line. This poem was written by a famous poet in Tang dynasty. It has been recited for a thousand of years. There could be many more other poems that are following this pattern or other green-purple patterns[1]. Taking figure 2 as an example, the regularity in green-purple patterns is interesting.

The first two lines share the following properties: first two characters have tones coming from one category; the following two characters have tones coming from a different category; the last character has a tone coming from the category that is same as that of the first two characters.

The last two lines share the following properties: the first three characters have tones coming from one category; the last two characters have tones coming from a different category.

Considering the first two lines as a pair and the last two lines as a pair, a property is shared by these two pairs: the second line in a pair is a flipped version of the first line in terms of color.

Above all, lots of information is contained in this green-purple pattern which makes the tone sequence partially predictable.

[1] Ping-Ze Patterns of Classical Chinese Poems
http://library.taiwanschoolnet.org/cyberfair2003/C0322500088/give/p203.htm

Question and Method

At this point, interesting questions arise: is an SRN able to discover tone grouping from tone sequences? Is SRN able to show clues for the underlying green-purple pattern?

Elman (1990) has shown in his ground breaking work that an SRN is able to discover lexical classes from word order. My problem in this project is highly similar to this one. In grammatical sentences, words appear one after another partially randomly. With memory, we may be able to predict that a noun is likely to follow a verb, but we may not be able to predict which noun will appear exactly. The underlying structure in Elman's problem are the templates for sentence generator. Similarly, the underlying structure in my problem is the green-purple pattern. At a certain time step, we are able to predict a green tone is more likely to happen than a purple tone given what has already been there, i.e., memory, however, we are not able to predict precisely which tone in a category is going to happen. Given the similarity between the problems, an SRN is suitable for this project.

If an SRN is able to discover lexical classes from grammatical sentences, it should be able to discover tone grouping from tone sequences. Elman (1990) did hierarchical clustering analysis of the network's internal representations of all words. Words that were similarly represented were clustered together. Similarly, clustering analysis could be done to show whether the network has learned the tone grouping.

With the XOR problem, the letter sequence problem and the word boundary problem, Elman (1990) showed that error pattern could associate very well with the temporal pattern that determines when the number or the letter is more predictable. Similarly, error pattern could show when a tone is more predictable, thus give us some insight of the underlying green-purple pattern.

Data

Tone sequences were simulated according to the green-purple pattern shown in Figure 2. First, the 4 by 5 green purple matrix were reshaped as a 20-bit vector with 0s and 1s representing green or purple blocks. Then each bit in this sequence was evaluated. 0 was converted to 1 or 2 randomly and equally likely. 1 was converted to 3 or 4 randomly and equally likely. An example of a generated tone sequence is shown below.

3 4 1 2 4 2 1 4 3 2 2 2 1 3 4 4 4 4 2 2

500 hundred tone sequences were generated in the same manner and were concatenated to form a long sequence. Then each number in this long sequence was evaluated and converted to a 4-bit vector in the following manner.

1→ [**1** 0 0 0]   2→ [0 **1** 0 0]   3→ [0 0 **1** 0]   4→ [0 0 0 **1**]

Only one bit of the vector is 1. All 4 vectors are orthogonal to each other, so there is no cues of how tones should be grouped in this representation.

Model

A simple recurrent neural network was built to solve the problem. The network has three layers: the input plus context layer, the output layer and the hidden layer. Since each tone is represented by a 4-bit vector, so there are 4 input units and 4 output units. The number of hidden units was decided to be fixed at 20. 10 and 30 were also tested. While 10 hidden units did not give meaningful error patterns, 20 hidden units and 30 hidden units gave similar results. Structure of the network is shown in Figure 3.

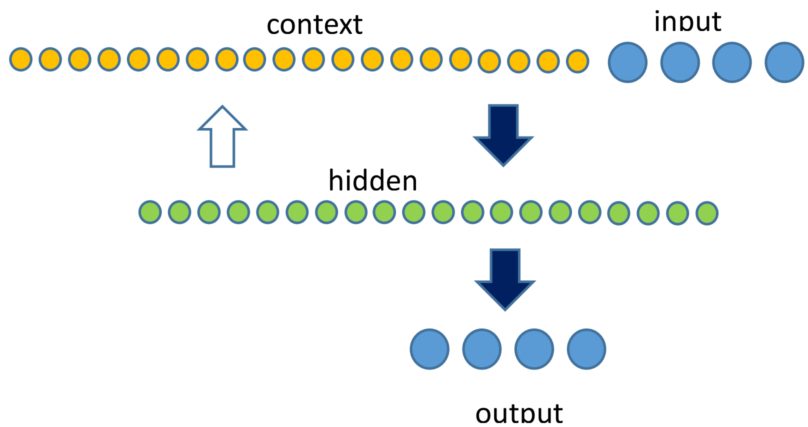

Figure 3. Structure of SRN employed in this project.

The learning rule is backpropagation through time. Learning rate was set to be 0.3. Training time was increased significantly when simulated annealing (SA) was added, but the performance almost did not change, so SA was removed later. No momentum term was involved. A sigmoid activation function was used. At each time step, the output of hidden units were copied to context units and became part of the input in the next time step.

The 4-bit vector was fed to the network one at a time. The task of the network is to predict the next 4-bit vector. The 10000 4-bit vectors were passed through the network 20 times.

Analysis and Results

Error at a certain time step was calculated as the root mean squared error (RMSE) from the 4-bit error vector. RMSE was averaged over 40 time steps for the last 10000 trials, which is shown in Figure 4 below. It is not surprising to find that a pattern repeats every 20 time steps, since our green-purple pattern repeats every 20 time steps.

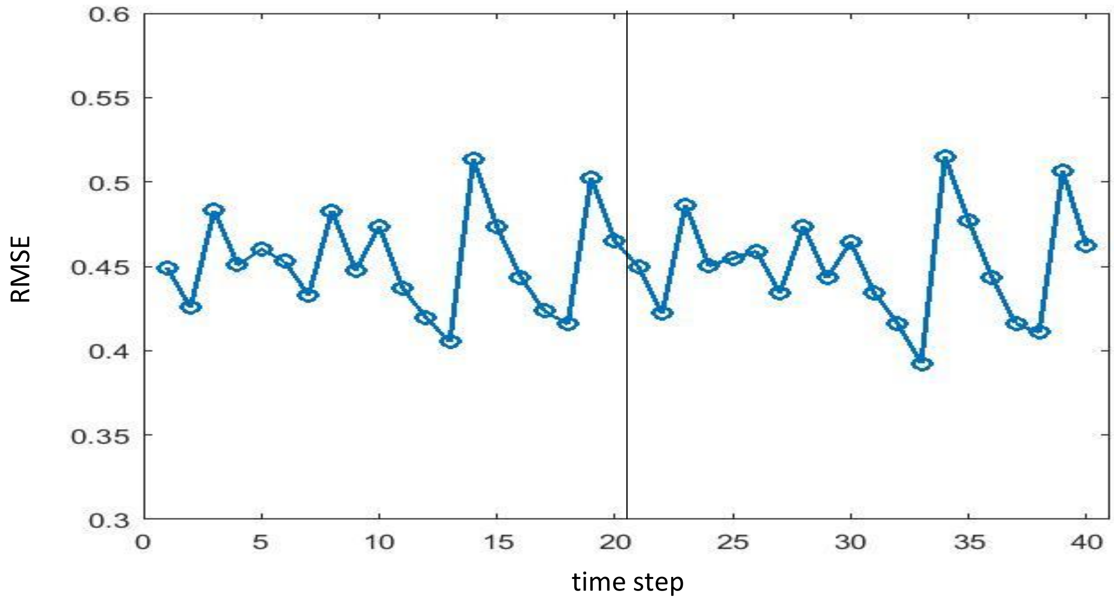

Figure 4. Averaged RMSE over 40 time steps in the last 10000 trials.

If we look closely at the error pattern for every 20 time steps and compare it with the green-purple pattern, it is obvious that the error pattern for the first five steps is similar to that for the second five steps. This similarity also holds for the third five steps and the last five steps. Within each 5 steps, the error goes down when the next tone is from the same category and goes up when the next tone is from a different category. Green and purple tones have similar RMSE at similar positions (e.g., step 1 and step 6, step 2 and step 7, etc.). The error pattern correlated very well with the green-purple pattern and the inner similarity between green tones and purples tones (i.e., green tones and purple tones are equally predictable.)

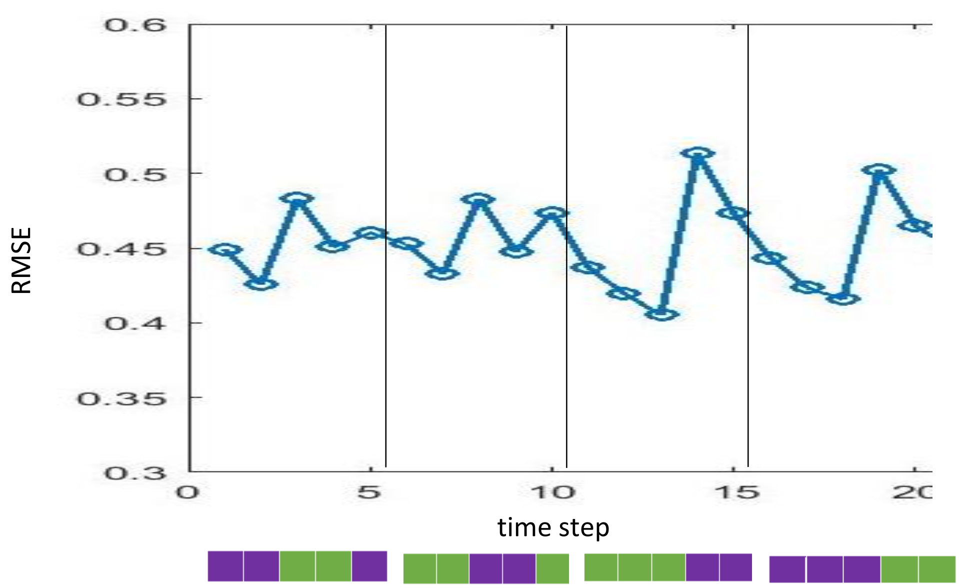

Figure 5. Averaged RMSE over 20 time steps.

To get the internal representation of each tone, the 10000 4-bit vectors were passed through the network one more time with the trained weights. In this process, weights were not updated. The output of hidden units at a certain time step is associated with a certain tone input. The output of hidden units for a certain tone were averaged over all contexts. In this way, each tone got a 20-bit vector as its internal representation. A k-means clustering analysis was performed (k = 2) over these four vectors. The result shows that the network learned that tone 1 and tone 2 should be grouped together while tone 3 and tone 4 should be grouped together.

cluster =

1	[1]	0.9997	0.9994	0.9999	0.0006	0.9993	0.9994	0.9102	0.9996	0.9993	0.9993	0.9994	0.9995	0.0482	0.9994	0.4978	0.9991	0.9996	0.9995	0.9996	0.9986
1	[2]	0.9995	0.9994	0.9997	0.0526	0.9996	0.9992	0.8991	0.9996	0.9984	0.9991	0.9997	0.9997	0.0460	0.9997	0.5312	0.9996	0.9995	0.9995	0.9997	0.9986
2	[3]	0.9996	0.9997	0.9995	0.0085	0.9995	0.9996	0.9975	0.9998	0.9992	0.9992	0.9998	0.9996	0.0085	0.9998	0.9455	0.9997	0.9996	0.9996	0.9998	0.9989
2	[4]	0.9994	0.9996	0.9994	0.0152	0.9994	0.9994	0.9935	0.9997	0.9993	0.9996	0.9994	0.9993	0.3003	0.9994	0.9205	0.9995	0.9990	0.9995	0.9995	0.9988

Figure 6. Clustering analysis result of the internal representations.

What if the randomness is reduced for green tones but kept same for purple tones? To evaluate if the error pattern is sensitive to the amount of randomness, instead of having equal probability (0.5), tone 1 was set to appear 70 percent of the time while tone 2 was set to appear only 30 percent of the time. This time, 5000 4-bit vectors were passed through the network 20 times. The error pattern in Figure 7 indicates the green-purple pattern is still repeating every 20 time steps. However, within each 20 steps, we see the pattern for first 5 steps is similar but different to that of the second 5 steps. This holds for the third 5 steps and the forth 5 steps. A detailed comparison for the third 5 and forth 5 steps is shown in Figure 8.

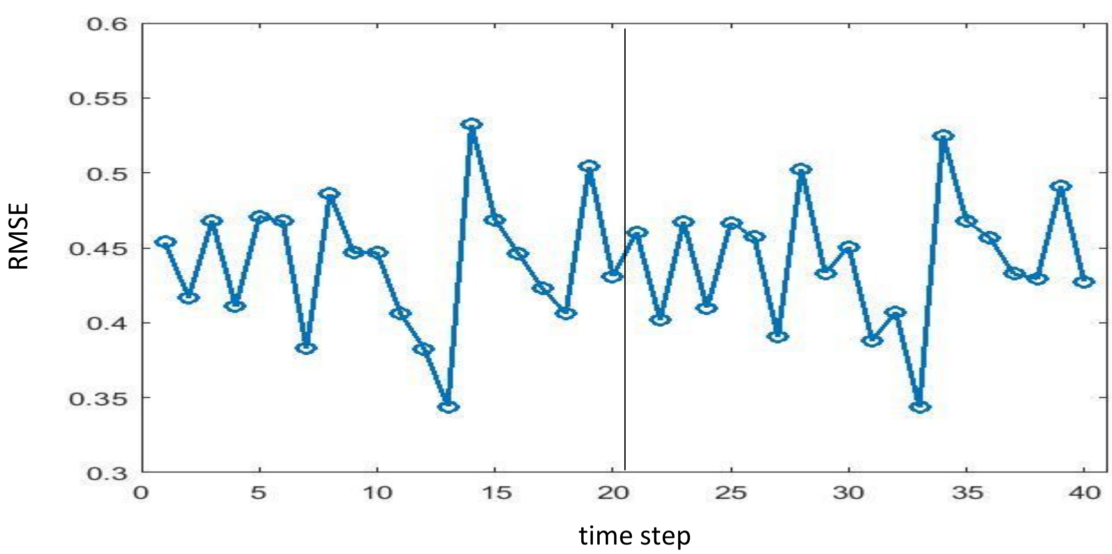

Figure 7. Error pattern when randomness was reduced for green tones.

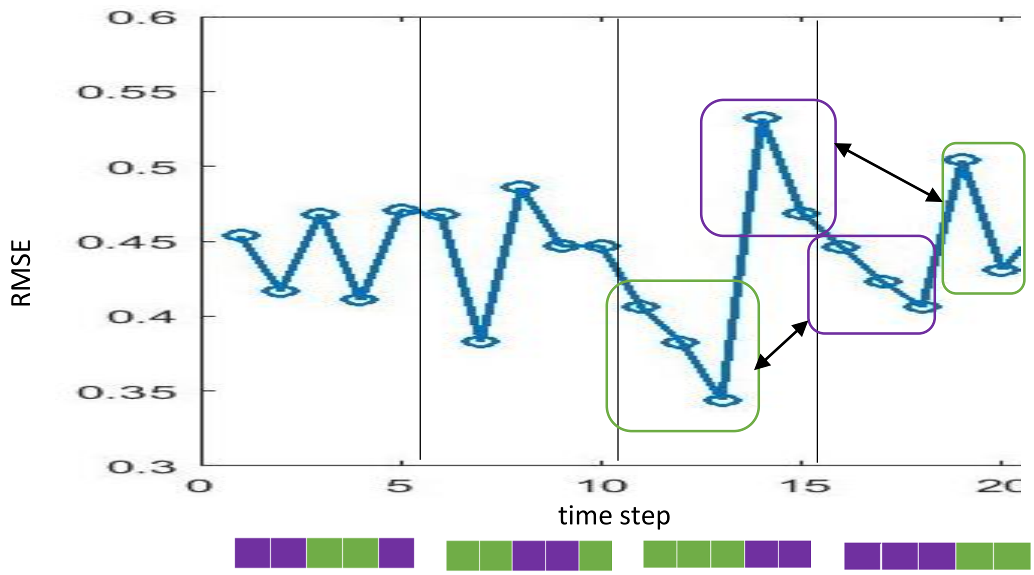

Figure 8. Averaged RMSE for 20 time steps when randomness was reduced for green tones.

Interestingly, green tones have lower RMSE than purple tones at similar positions (e.g., 11,12,13 VS 16,17,18). The error pattern is very sensitive to the amount of randomness.

This project shows that recurrent neural networks may be suitable for finding out tone patterns in Chinese poems. However, only a limited amount of simulated data were presented here. More complicated patterns could be tested. Noise could be added. Data from real poems could be tested to find out tone patterns of poems of certain styles.

(2029 words)

References

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, *14*(2), 179-211.

Thrasher, A. R. (Ed.). (2016). *Qupai in Chinese Music: Melodic Models in Form and Practice*. Routledge.