

# Jazz Melody Generation from Recurrent Network Learning of Several Human Melodies

Judy Franklin

Computer Science Department  
Smith College  
Northampton, MA 01063

## Abstract

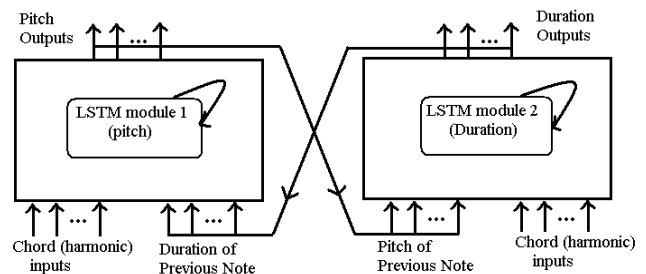
Recurrent (neural) networks have been deployed as models for learning musical processes, by computational scientists who study processes such as dynamic systems. Over time, more intricate music has been learned as the state of the art in recurrent networks improves. One particular recurrent network, the Long Short-Term Memory (LSTM) network shows promise as a module that can learn long songs, and generate new songs. We are experimenting with using two LSTM modules to cooperatively learn several human melodies, based on the songs' harmonic structures, and the feedback inherent in the network. We show that these networks can learn to reproduce four human melodies. We then introduce two harmonizations, constructed by us, that are given to the learned networks. i.e. we supply a reharmonization of the song structure, so as to generate new songs. We describe the reharmonizations, and show the new melodies that result. We also use a different harmonic structure from an existing jazz song not in the training set, to generate a new melody.

## LSTM Networks as Modules in a Music Learning System

Recurrent neural networks are artificial neural networks that have connections from the outputs of some or all of the network's nonlinear processing units back to some or all of the inputs. These networks are trained by repeatedly presenting inputs and target outputs and iteratively adjusting the connecting weights so as to minimize some error measure. The advantage of recurrent neural networks is that outputs are functions of previous states of the network, and sequential relationships can be learned. However, this very facet causes the weight update equations to be much more complex than simple non-recurrent neural networks, to correct for using erroneous outputs in previous time steps. And it is difficult to design a stable network that can learn long sequences. Yet, this is necessary for musical learning systems.

Copyright © 2005, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

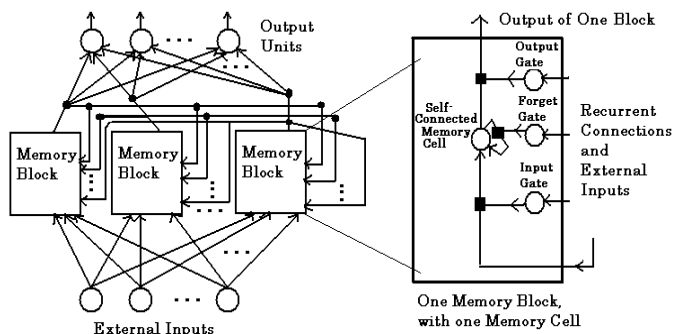
In recent publications (to be cited in final paper), we have shown that a particular recurrent neural network, the long short-term memory network (LSTM), can learn to distinguish musical pitch sequences, and can learn long songs. Here we present a two-module LSTM system that can learn both pitch and duration of notes in several long songs, and can subsequently be used to generate new songs. While we have developed systems before based on similar ideas, the LSTM-based system is much more precise and stable, and can learn much longer songs. Figure 1 shows our two-module LSTM configuration for learning songs. This configuration is inspired by Mozer's (1994) CONCERT system that uses one recurrent network, but with two sets of outputs, one for pitch and one for duration. It is also inspired by Eck and Schmidhuber's (2002) use of two LSTM networks for blues music learning, in which one network learns chords and one learns pitches; duration is determined by how many network iterations a single pitch remains on the output.



**Figure 1. The two-module LSTM system. One LSTM module learns pitches. The other learns note durations. The recurrence in each LSTM network is shown internally.**

Each LSTM module contains an LSTM neural network. An LSTM neural network is a kind of recurrent neural network with conventional input and output units, but with an unconventional, recurrent, hidden layer of memory blocks (Hochreiter and Schmidhuber 1997, Gers et al. 2000). Each memory block contains several units (see Figure 2). First there are one or more self-recurrent linear memory cells per block. Second, each block contains three gating units that are typical sigmoid units, but are used in

the unusual way of controlling access to the memory cells. One gate learns to control when the cells' outputs are passed out of the block, one learns to control when inputs are allowed to pass in to the cells, and a third one learns when it is appropriate to reset the memory cells. LSTM's designers were driven by the desire to design a network that could overcome the vanishing gradient problem (Hochreiter et al. 2001). Over time, as gradient information is passed backward to update weights whose values affect later outputs, the error/gradient information is continually decreased by weight update scalar values that are typically less than one. Because of this, the gradient vanishes. Yet the presence of an input value way back in time may be the best predictor of a value far forward in time. LSTM offers a mechanism where linear units can store important data without degradation, for long periods of time, in order to decrease vanishing gradient effects.



**Figure 2.** An LSTM network on the left, and a more detailed enlargement of a memory block that contains one memory cell, and the three gating units.

As shown in Figure 1, one LSTM network learns to reproduce the pitches of one or more songs, and a second one learns to output the corresponding durations. The dual system contains recurrence in three places: the inter-recurrence at the network level, the recurrence of the hidden layer of memory blocks, and the self-recurrence of each memory cell. We showed in previous work (to be cited) that a similar system, with two LSTM networks that are not inter-recurrent, can learn both a human rendition of the song “Afro Blue”, as well as a score-based version. The two networks can also learn Afro Blue with and without the chordal structure given as input. To expand the system to be able to generate music, we tie the pitch and duration networks together, so each network receives the outputs of the other networks, for the previous note. We also present the harmonic structure corresponding to the example song to be learned to each network's input units. These inputs are in the form of the chord over which the current melody notes are being played. A small amount of beat information is given to the networks. One input has the value of 1 if the beginning of a new measure has passed. Finally, each song

to be learned is assigned a number. Another set of inputs not shown in the diagram are a binary encoding of the number of the song being learned (one binary input for each song, which is 1 only if that song is the current training example).

## The Four Songs

The four songs learned by the two-module system are “Blue Bossa”, “Summertime”, “Watermelon Man”, and “Cantaloupe Island.” The four songs are presented here, each as a musical score of a human rendition of the melody, also showing the chord structure. Figures 3 and 4 show the songs Summertime and Watermelon Man. The human renditions were obtained from MIDI files found on the web. The chords for each song are from those provided in (Aebersold 2000).

### Summertime

The musical score for 'Summertime' is presented in four staves. The top staff shows the melody in treble clef with chords: A7alt, Dm, Dm, Dm. The second staff shows the bass line with chords: Dm, Gm, Gm, Em7b5. The third staff continues the melody with chords: A7alt, Dm, Dm, Dm. The bottom staff continues the bass line with chords: Gm, C7, F, Em7b5, A7alt, Dm.

**Figure 3.** Summertime score, showing human rendition and harmonic structure.

### Watermelon Man

The musical score for 'Watermelon Man' is presented in four staves. The top staff shows the melody in treble clef with chords: F7, F7, F7, F7. The second staff shows the bass line with chords: Bb7, Bb7, F7, F7. The third staff continues the melody with chords: C7, Bb7, C7, Bb7. The bottom staff continues the bass line with chords: C7, Bb7, F7, F7.

**Figure 4.** Watermelon Man score, showing human rendition and harmonic structure.

Figures 5 and 6 show the other two of the four songs, Blue Bossa, and Cantaloupe Island. Each song has a different harmonic structure, although there is some overlap in the individual chords that appear. Each song is in 4/4 time, with four beats per bar, and each has 16 bars. Three of the songs have lead-in notes before the first bar. The presentation of the songs as examples includes one lead-in measure so as to include the lead-in notes.

This next section describes the representation of pitch and duration, as well as the learning parameters for the experiments.

### Blue Bossa



Figure 4. Blue Bossa score, showing human rendition and harmonic structure.

### Cantaloupe Island

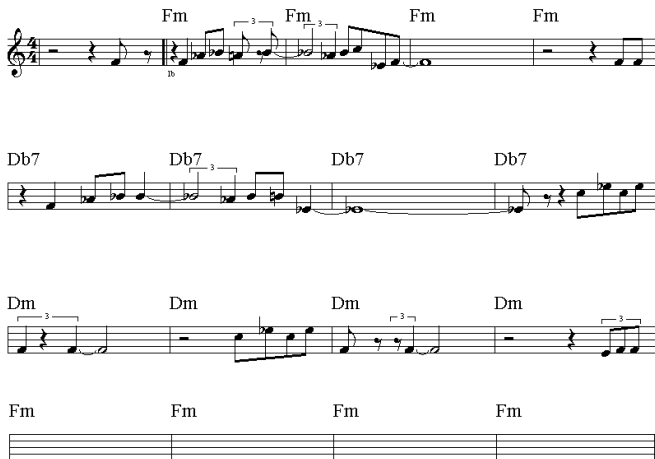


Figure 5. Cantaloupe Island score, showing human rendition and harmonic structure. Notice that no notes are played in the last four bars of the song.

## Experimental Details

### Pitch Representation

The pitch of the note corresponds to the note's semi-tone, from Western tonal music. Pitch must be represented numerically, and there are many ways to do this from a musicological point of view (Selfridge-Field 1998). Since our melody sources are MIDI-based, we often think of pitches as having an integer value, one value for each semi-tone, with 60 representing middle-C. But the pitch must be represented in a way that will enable a recurrent network to easily distinguish pitches. We have developed one such representation called Circles of Thirds. We have experimented with this representation on various musical tasks, with successful results (citations will be made in the final paper), and have compared it to others such as those found in (Todd 1991, Mozer 1994, Eck and Schmidhuber 2002).

Figure 6 shows the three circles of major thirds, a major third being 4 half steps, and the four circles of minor thirds, a minor third being 3 half steps.

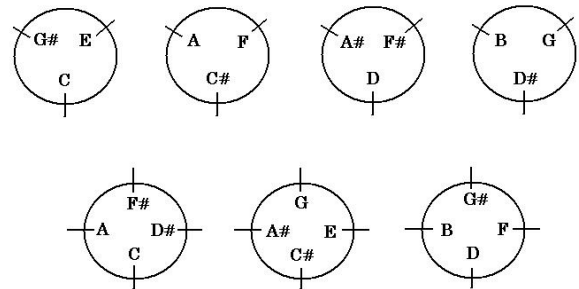


Figure 6. At top, circles of major thirds. At bottom, circles of minor thirds. A pitch is uniquely represented via these circles, ignoring octaves.

The representation consists of 7 bits. The first 4 indicate the circle of major thirds in which the pitch lies, and the second 3, the circle of minor thirds. The number of the circle the pitch lies in is encoded. C's representation is 1000100, indicating major circle 1 and minor circle 1, and D's is 0010001, indicating major circle 3, and minor circle 3. D# is 0001100. Because the 7<sup>th</sup> chord tone is so important to jazz, our chords are the triad plus 7<sup>th</sup>. In using Circles of Thirds to represent chords, we could represent chords as 4 separate pitches, each with 7 bits for a total of 28 bits. However, it would be left up to the network to learn the relationship between chord tones. We borrowed from Laden and Keefe's (1991) research on overlapping chord tones as well as Mozer's (1994) more concise representation for chords. The result is a

representation for each chord that is 7 values. Each value is the sum of the number of on bits for each note in the chord. For example, a C<sup>7</sup> chord in a 28 bit Circles of Thirds representation is

<u>1000100</u>	<u>1000010</u>	<u>0001010</u>	<u>0010010</u>
C	E	G	B-flat

The overlapping representation is:

1000100	(C)
1000010	(E)
0001010	(G)
<u>+0010010</u>	(B-flat)
2011130	(C <sup>7</sup> chord)

### Duration Representation

We have also experimented with duration representations (to be cited in final paper). In our system, the entire note duration is the output of one LSTM module on one iteration. In our Modular Duration representation, beat length is divided by 96 giving 96 clicks per quarter note, 48 per eighth note, 32 per eighth note triplet note, etc. We can represent triplets and swing, and duration variations that occur in human MIDI performance (Thomson 2004), a step toward interpreting expressive MIDI performances. Our representation is a set of 16 binary values. Given a duration value, dur, the 16<sup>th</sup> bit is 1 if  $dur/384 \geq 1$ , where  $384 = 96 * 4$ , is the duration in clicks of a whole note. Then the 15<sup>th</sup> bit is 1 if  $(dur \% 384) / 288 \geq 1$ . In other words if the remainder after dividing by 384 and then dividing by 288 is greater than or equal to 1. The 14<sup>th</sup> bit is 1 if  $(dur \% 384 \% 288) / 192 \geq 1$ . The modulo dividers are 384, 288, 192, 96, 64, 48, 32, 24, 16, 12, 8, 6, 4, 3, 2, and 1, corresponding to whole note, dotted half, half, quarter, dotted eighth, eighth, eighth triplet, sixteenth, sixteenth triplet, thirty-second, and then 6, 4, 3, 2, and 1 for completeness. Any duration that exactly matches, in clicks, one of these standard score-notated durations can be represented, as can combinations of them, or human-performed approximations to them.

Two example durations (in clicks) from Summertime are 86 and 202. The duration 86 is  $64+16+6$ , represented as 0000100010010000, and the representation of 202 is  $192+8+2$ , or 0010000000100010.

### Experimental Results

We base the choice of parameters for the two LSTM modules on those values that worked best in the past on specific musical tasks and on the learning of the pitch and duration of the melody of Afro Blue. Consequently, both of the two LSTM modules contain 20 memory blocks, with four cells each. The set of four songs is presented for 15000 epochs. The two-module network learns to reproduce the four songs *exactly*, with a learning rate of .15 on the output units, and a slower rate of .05 on all other units. A larger rate on the output units produces consistently stable and accurate results in our previous experiments as well. Once the four songs were learned, we

began experimentation with generating new melodies. In this network configuration, a straightforward way to do this is to give the networks a whole new chordal structure. We keep the inter-recurrent connections and set the four song inputs all equal to one (all on).

#### Melody Generated on Complex Chords



**Figure 7. The melody generated by the dual-network system, over a complex chord structure.**

We show melodies that are generated over three different harmonic structures, in Figures 7-9. The figures also show the harmonic structure as before. One bar of a pick-up or lead-in chord is given in each chord structure, since three out of four training songs had lead-in notes. Figure 7 shows a melody generated over a fairly complex harmonic structure that we derived from the structures of the four learned songs. There is a new chord in every bar except for the occurrence of Fminor two bars in a row in the second line. The melody depicted is a close approximation of the actual melody output by the networks. The approximation is made by the software used, Band-in-a-Box (PG Music 2004), to enter in the chords, to import the MIDI file, and to generate the scores as shown in the figures.

While there are a couple of notes out of place, such as the initial A on the G7alt chord in the lead-in bar, and the F# on the G7alt four bars later, the melody notes are derived from the scales one might associate with the chords when improvising, and the rhythm is quite reasonable.



## References

Aebersold, J. 2000. *Maiden Voyage (Vol 54)*. New Albany, IN : Jamey Aebersold.

Eck, D. and Schmidhuber, J. 2002. Learning the Long-Term Structure of the Blues. *Proceedings of the 2002 International Conference on Artificial Neural Networks (ICANN)*. 284-289.

Engelmore, R., and Morgan, A. eds. 1986. *Blackboard Systems*. Reading, Mass.: Addison-Wesley.

Gers, F. A., Schmidhuber, J. and Cummins, F. 2000. Learning to forget: Continual prediction with lstm. **Neural Computation** 12(10): 2451-2471.

Griffith, N. and Todd, P. 1999. *Musical Networks: Parallel Distributed Perception and Performance*. MIT Press, Cambridge MA.

Hochreiter, S. and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735-1780.

Hochreiter, S., Bengio, Y., Frasconi, P., and Schmidhuber, J. 2001. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. *A Field Guide to Dynamical Recurrent Networks*. IEEE Press, New York, NY.

Laden, B., & Keefe, D.H., 1991. The Representation of Pitch in a Neural Net Model of Chord Classification. *Music and Connectionism*, Todd, P.M., Loy, E.D., eds., Cambridge, MA. MIT Press

Mozer, M. C., 1994. Neural Network Music Composition by Prediction: Exploring the Benefits of Psychophysical Constraints and Multiscale Processing. *Connection Science*, 6, 247-280

Todd, P. M., & Loy, E. D., 1991. *Music and Connectionism*, Cambridge, MA: MIT Press

Selfridge-Field, E. 1998. Conceptual and Representational Issues in Melodic Comparison. *In Melodic Similarity. Concepts, Procedures, and Applications. Computing in Musicology 11*. Hewlett, W. and Selfridge-Field, E., eds. Cambridge MA, MIT Press.

Todd, P. M., 1991. A Connectionist Approach to Algorithmic Composition, *Music and Connectionism*, eds.: Todd, P.M. and Loy, E. D., Cambridge, MA, MIT Press