Chapter 2 (DRAFT)

Hugh Zabriskie

3 December 2015

1 Literature Review

1.1 Purpose of the lit review

1. To inform the reader of the most important research needed to understand the research question.
2. To give me credibility as someone who knows what they are talking about.
3. To be organized so that the the review leads the reader to a "gap" or "conflict" in the literature.

1.2 Previous Work

Hild, Feulner, and Menzel (1992) presented the first effective neural network approach for harmonizing chorales, generation harmonizations on the level of an "improvising organist" (p. 272). The task decomposed into three subtasks, each learned by a neural net. A harmonic skeleton is first created by sweeping through the chorale melody and determining a harmony for each beat, where harmony is represented by a unique figured bass notation. For each time step t, the network takes an input a window of information, including the harmonies chosen in the interval$[t-3, t-1]$ and the melody pitches in the interval $[t-1, t+1]$. The resulting harmonic skeleton is passed to another network to generate a chord skeleton,

which selects the inner voices based on the input figured bass, and a final network adds ornamental eighth notes to restore passing tones between chords. A weakness in HARMONET is the addition of external "chorale constraints" (Hild, Feulner, and Menzel, 1992, p. 271) that are used to construct the chord skeleton, introducing a potential distortion in the network's learning process. Allan and Williams (2005) removed these constraints when they implemented this approach using Hidden Markov Models (HMMs).

Substantial work has been done in the field of music generation using RNNs and LSTMs that demonstrates their ability to learn complicated musical structures and relate temporally distant events. Toiviainen (1995) presents a neural network that generates a jazz bebop melody over series of chord changes. The network achieves melodic continuity by using a "target-note technique", where the end of the melodic pattern (the target notes) over the previous harmony is used to generate the melody over the current harmony. Eck and Schmidhuber (2002) used LSTMs to compose chords and a suitable melody over a 12-bar blues form. The authors improved on the results of Mozer (1994), who used RNNs to compose melodies with chordal accompaniment but found the resulting music lacked larger thematic and phrase structure. They attribute the cause of Mozer's concern to the "vanishing gradients" problem (described briefly in chapter 1) and use LSTMs to first generate the chordal structure and use that as a input to the LSTMs that generates a melody. Franklin (2006) uses two inter-recurrent LSTMs to compose jazz melodies over a chord progression, training on a dataset of well-known jazz standard melodies. Goel, Vohra, and Sahoo (2014) demonstrated the successful application of their modified LSTM architecture to the task of polyphonic music generation over a collection of musical datasets that includes Bach's chorale harmonizations.

Work has also been done to determine with network are most favorable to the

task of harmonization. A large-scale analysis of eight LSTM networks on the task of chorale harmonization was conducted by Greff et al. (2015) to determine which modifications represent a significant improvement in architecture. The chorale dataset was the popular MIDI dataset created by Allan and Williams (2005), and the evaluation metric was negative log likelihood. Each modification was simple enough to be isolated, such as the removal of a gate in the LSTM memory cell or full gate recurrence (FGR), which adds a recurrent connection at each gate. Interestingly, they discovered that the none of the modified architecture represented significant improvements over the "vanilla", unmodified LSTM (Greff et al., 2015, p. 7). Instead, the choice of hyperparameters, such as learning rate and hidden layer size, were determined to be the most crucial factors. Moreover, each hyperparameter could be adjusted independently since the interactions between them was measured to be relatively small. Goel, Vohra, and Sahoo (2014), however, did find a substantial improvement in their LSTM architecture that incorporates a Deep Belief Network (DBN), which learns to extract a deep hierarchical representation of the music data.

2 Precursor: Chorale Harmonization Implementation

I will first replicate results on recent chorale harmonization using LSTMs to create a baseline model for the harmonization task. The chorale data used in this paper comes from the corpus provided with MUSIC21, maintained by Professor Michael Scott Cuthbert at MIDI and originally obtained from the MIDI archive at Yale University. Recent papers (Greff et al., 2015; Allan and Williams, 2005; Goel, Vohra, and Sahoo, 2014) have depended on a chorale data set encoded in MIDI, but this representation fails to distinguish between the melody and the harmonization beneath. The data used here correctly isolates each voice within the chorale, restoring

information about the linearity of the voices and events such as voice crossing.

2.1 Preprocessing

In order to create a numerical representation of the chorale data, I used iterated over the 348 four-voice chorales and generated a feature space for each of the following features.

1. Pitch, encoded in MIDI values. A search across all chorales indicated that each voice had a well-defined pitch range, verified by Madsen (2002):
    - *soprano*: [60, 81]
    - *alto*: [53, 74]
    - *tenor*: [48, 69]
    - *bass*: [36, 64]
2. Beat strength, where 1 represents the downbeat within a measure.
3. Fermata present - a binary feature to indicate points of cadential resolution.
4. Number of quarter-note beats until the next fermata.
5. Number of beats until the end of the chorale.
6. Time signature (almost exclusively '3/4' and '4/4').
7. Key signature, represented the number of sharps or flats.
8. Key mode, either major or minor.

The Python script used to preprocess the chorales (see wrangle.py in Appendix B) [1] extracts the above features from each time step of each chorale, and stores the generated training and tests in an HDF5 file. The output is then fed to a Lua script that uses Torch, a scientific computing framework with machine learning libraries, to construct the network and performed classified learning.

[1] There will be an Appendix B that contains the most important code for the project.

References

Allan, Moray and Christopher KI Williams (2005). "Harmonising chorales by probabilistic inference". In: *Advances in neural information processing systems* 17, pp. 25–32.

Eck, Douglas and Juergen Schmidhuber (2002). "A first look at music composition using lstm recurrent neural networks". In: *Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale*.

Franklin, Judy A (2006). "Jazz melody generation using recurrent networks and reinforcement learning". In: *International Journal on Artificial Intelligence Tools* 15.04, pp. 623–650.

Goel, Kratarth, Raunaq Vohra, and JK Sahoo (2014). "Polyphonic Music Generation by Modeling Temporal Dependencies Using a RNN-DBN". In: *Artificial Neural Networks and Machine Learning–ICANN 2014*. Springer, pp. 217–224.

Greff, Klaus et al. (2015). "LSTM: A Search Space Odyssey". In: *arXiv preprint arXiv:1503.04069*. URL: http://adsabs.harvard.edu/abs/2015arXiv150304069G.

Hild, Hermann, Johannes Feulner, and Wolfram Menzel (1992). "HARMONET: A neural net for harmonizing chorales in the style of JS Bach". In: *Advances in Neural Information Processing Systems*, pp. 267–274.

Mozer, Michael C (1994). "Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multi-scale processing". In: *Connection Science* 6.2-3, pp. 247–280.

Toiviainen, Petri (1995). "Modeling the Target-Note Technique of Bebop-Style Jazz Improvisation: An Artificial Neural Network Approach". English. In: *Music Perception: An Interdisciplinary Journal* 12.4, pp. 399–413. ISSN: 07307829. URL: http://www.jstor.org/stable/40285674.