A Generative Model of Pulse Percept: Analyzing Performances of Free Jazz Drumming using Dynamic Beat Tracking and Recurrent Neural Networks

Nolan Lem

Introduction

The establishment of the *pulse percept,* the basic unit of perceived rhythm that constitutes a musical beat, is critical to the way in which we listen, engage, and perform with music. Fundamental to this notion of pulse is the way in which it orientates cognitive processes involved in anticipation, expectation, and arousal. Taking cues from connectionist theories related to musical structure, pulse can be construed to be an organizational unit from which we derive emergent orders of temporal structure (Jackendoff & Lerhdal 1983). Pulse extraction is the means through which we are able to manage time-dependent events, namely those that occur with some level of regularity.

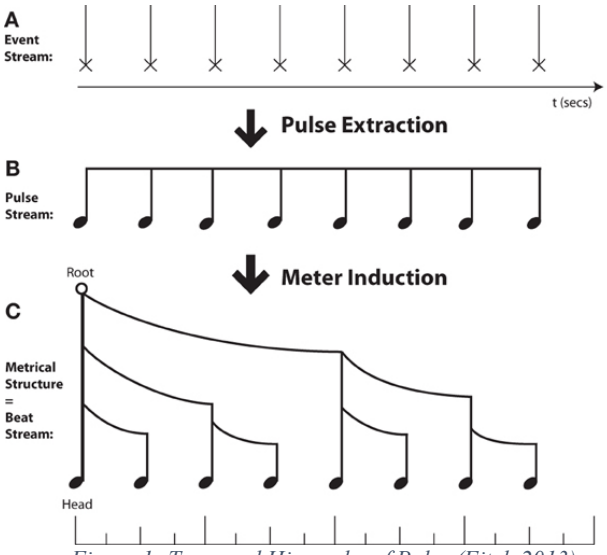

Figure 1: Temporal Hierarchy of Pulse (Fitch 2013)

For most music that adheres to fixed tempo (e.g. dance music), pulse can be thought of as the beat that is most perceptually salient to the listener. For instance, several studies have used beat tapping data as criteria from which to understand the perception of pulse (Ellis 2007, Davis et al. 2009, Holzapfel 2012) within a listener. One confounding principle in pulse perception is the notion of the missing 'fundamental': some musical stimuli may induce a pulse percept that is not physically present in the acoustic signal itself (Large et al. 2015). Implicit in this formulation is the idea that we are not necessarily conscious of the process of becoming entrained to an incoming pulse stream. How do rhythmic stimuli induce percepts of pulse within the listener in the absence of any strictly periodic acoustic information? Taken from a different perspective, how is the pulse percept used by a performer to organize time in the course of a performance of music?

This study looks at the ways in which recurrent neural networks can encode pulse from music that is only marginally periodic with respect to a beat. For this, I refer to this characterization of pulse as *quasi-periodic* which I define to be music that contains implicit references to locally isochronous temporal events but does not adhere to a fixed tempo. Based upon this definition, improvisative music is well suited to the definition just described. This study aims to derive a novel representation of pulse onsets from analyzing audio from solo jazz drumming performances.

Much current research is exploring the ways in which neural networks can be applied to create artificially intelligent generative music. Previous research in the creative application of musical machine learning includes style transfer (Ulyanov 2016), music information retrieval (MIR), and generative/interactive models capable of

improvising with performers in real time (Eck 2007). While some beat detection algorithms leveraging neural networks have focused on beat induction in fixed tempo music (see Lambert et al. 2015, Eck & Schmidhuber 2002), I'm interested in applying network learning procedures to quasi-periodic improvisatory music as a way to learn about how rhythmic events are derived from pulse percepts. In doing so, I hope to shed light on the way in which a performer organizes time in relation to a perceived pulse. As such this project is an attempt to reveal any beat-oriented organizational structure that might inform a jazz drummer's sense of pulse during the course of an improvisation.

This study's aim is twofold: to evaluate the extent to which recurrent neural networks are capable of incorporating learned pulse representations from its input and to establish a generative model that encompasses learned, rhythmic aspects of a specific drummer's style. From this I want to explore how this model might be generalizable to other performances of jazz drumming as an analytical and musicological device.

Methods

This study looks at a singular performance of jazz drums from the album, *'nommo'*, by Milford Graves and Don Pullen recorded in 1967[1]. This particular album was chosen because of the stylistic performance of the drummer, Milford Graves, which seemed to adhere to a style of playing that fit the prior description of 'quasi-periodic'. It was also deemed to be representative of free jazz insofar as the study's findings might be generalizable with respect to genre.

This study examined two types of recurrent neural networks (RNNs): stacked Long-Short Term Memory (LSTM) RNNs first developed by Hochreiter and Schmidhuber in 1997 and Clockwork (CW) RNNs developed by Koutník in 2014. LSTM-RNNs have been shown to be effective in handling temporally-dependent sequences such as time-series data and have been used extensively to create generative models of language (Karpathy 2015). However CW-RNNs have been shown to outperform LSTM-RNNs in spoken word recognition and handwriting recognition tasks (Koutník et al. 2014). Recently machine listening in the context of music have shown how CW-RNNs outperform LSTM-RNNs in real-time music composition tasks via generative models (Sidor & Jack 2016). In meter perception studies, gradient frequency neural networks (Large et al. 2010) have been introduced as input to LSTM-RNNs to study their ability to track periodic changes of tempo and learn long-term temporal structure from an audio signal (Lambert et al. 2014).

The approach highlighted in this study is similar to word generation models insofar as it treats sequences of beat onsets as *temporally dependent* given the context of the auditory scene in much the same way that words are treated as *spatially dependent* on the context of the word and the sentence. This study aims to explore the extent to which underlying rhythmic structure can be learned from these RNNs that have shown success in uncovering organizational structure in sequences of data.

The training set is comprised of eighty, ten second samples of solo drum recordings from the free jazz album, 'nommo', performed by Milton Graves. Each ten second sample is analyzed for the beat onset information

[1] Graves, M. & Pullen, D. (1967). *nommo* [compact disc]. New Haven, NJ: SRP Records.

(described in the next section) and provided as input into the LSTM and CW-RNNs. After training, the models can be seeded with data from the test set to produce generative sequences of beat onsets.

Spectral Beat Onsets and Local Tempo Estimates

This model is dependent on beat information being appropriately encoded from the raw audio signal so that the RNN can learn to predict and generate beats. More specifically, the beat tracking algorithm used in this network employs a dynamic programming algorithm that balances spectral envelope onsets with global tempo estimates (Ellis 2007). In this way, both locally specific onsets (spectral envelope) are paired with temporally regular onsets (global tempo estimates) to create representation of rhythm that contains both periodic and non-periodic information pertaining to the beat[2]. The beat tracking data was collected using the Librosa[3] audio analysis python library. This next section provides an overview of the Ellis style[4] beat tracking methods used in this study.

Spectral Beat Onsets

The spectral beat onsets are calculated using a basic perceptual model that uses peak picking from Short Time Fourier Transform (STFT) frames (hopsize=512 samples) and an auditory representation mapping to 40 Mel bands via a weighted summing of the spectrogram values (Ellis 2005). The algorithm provides a representation of a one-dimensional onset strength envelope as a function of time that corresponds to the proportional increase energy summed across auditory frequency bands [5]. This approach has shown to work well for a variety of timbres in different musical contexts (Bello et al. 2005). Figure 2 and Figure 3 provides example plots (in frames and samples respectively) of the spectral strength envelope corresponding to two different ten second audio clips used in the training set.

[2] This approach would be less useful for music that adheres to a fixed tempo for instance. Because of the nature of free jazz drums (quasi-periodic) where an objective 'beat' cannot be easily established, it seemed critical to include both onset measures (spectral onsets and global tempo) as input the neural net so that the network might learn to find patterns between the temporally periodic onsets inherent to the global tempo and the more stochastic onset information specified by the spectral envelopes.

[3] https://github.com/librosa/librosa

[4] For a more comprehensive documentation of this beat tracking algorithm please see '*Dynamic Beat Tracking'* (Ellis 2007).

[5] This Mel Spectrogram is converted to dB and first-order differences across time are calculated for each band. The negative values are rectified to zero and the positive differences are summed across frequency bands. This signal is passed through a high-pass filter with a fc around 0.4 Hz to make it local mean and smoothed with a Gaussian envelope.

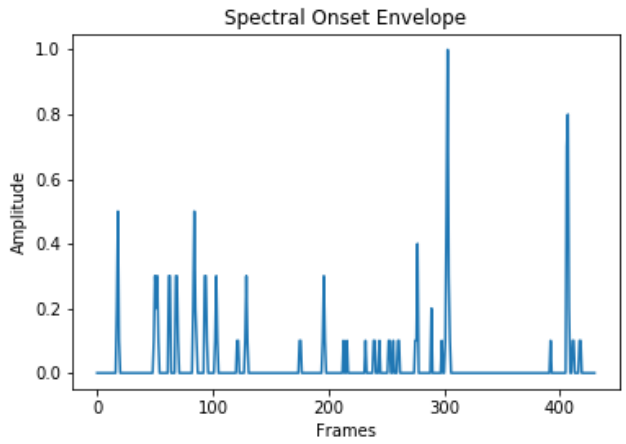

Figure 2: Spectral Onset Envelope (actual representation)

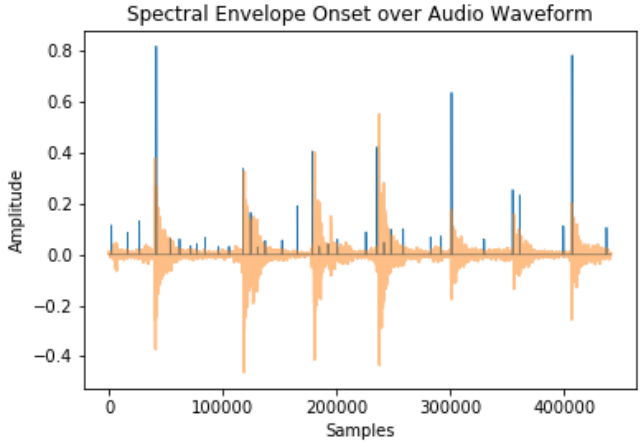

Figure 3: Spectral Onset Envelope with Audio Waveform

Global Tempo Estimates

To derive the global tempo estimates, a cost transition objective function, C($\{t_i\}$) is defined in [1].

$$C(\{t_i\}) = \sum_{i=1}^{N} O(t_i) + \alpha \sum_{i=2}^{N} F(t_i - t_{i-1}, \tau_p) \qquad [1]$$

Here, $\{t_i\}$ is the sequence of beat instances found by the tracker, O(t) is the spectral onset envelope from the audio, F(t) is a consistency function that mitigates inter-beat interval t with an ideal beat spacing defined by the target tempo (τ_p), α is a weighting to balance the two terms above. F(t) is defined in [2].

$$F(\Delta t, \tau) = -(log \frac{\Delta t}{\tau})^2 \qquad [2]$$

The objective function can derive a best-scoring time sequences by iterating recursively to calculate the best possible score C*(t) (via Bellman) [3] and while recording the actual preceding beat time that gave this score as shown in P*(t) in [4].

$$C^*(t) = O(t) + \max_{\tau=0..t}\{\alpha F(t - \tau, \tau_p) + C^*(\tau)\} \qquad [3]$$

$$P^*(t) = arg \max_{\tau=0..t}\{\alpha F(t - \tau, \tau_p) + C^*(\tau)\} \qquad [4]$$

The procedure for finding the optimal beat times (maximizing the cost-transition function) is as follows

1.) Calculate C* and P* for every time starting from time 0.
2.) Look for the largest value of C* (typically toward the end of the sequence) to form final beat instance t_N.
3.) Recursively iterate via P* finding preceding beat time $t_{N-1} = P^*(t_N)$ and working backwards until reaching time 0.

The procedure searches the entire exponentially-sized set of all possible time sequences. In this study, this optimal beat sequence forms the 'periodic' local tempo estimates in frames (1 frame = 512 samples). Figure 4 shows an example of these local beat estimates as onsets over the course of an audio training example.

Note how the local beat estimates (blue vertical lines) are much more periodic in time relative to the spectral onsets (in red) which give priority to local events. This is due to the fact that they are estimating a tempo given information about both the spectral onsets and a global tempo estimate that emerges from the objective function just described. This example illustrates a single beat-onset vector that will eventually become parsed into windowed subsequences and fed into the neural networks described below.

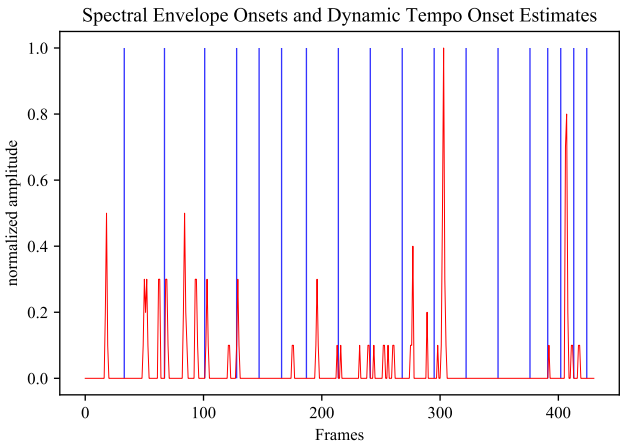

Figure 4: Global Tempo Estimate Onsets (Blue) and Spectral Envelope Onsets (red)

Armed with both the onset detection functions—the spectral onsets and the global beat onset estimates—this study combined both of these mid-level beat representations as input to the RNNs.

RNN Architecture

Training Set

The beat onsets described in the previous section are used as inputs to an LSTM and CW-RNN with two hidden layers (see architecture) and one fully-connected layer with a softmax output. The one-hundred training examples were sampled at 22050 Hz. The ten second input beat onset vectors were in terms of frames (≈ 431 frames per training example where 1frame = 512 samples ≈ 23ms) due to the beat onset algorithms using STFT with a hopsize of 512 with N=sr. This reduces the dimensionality of the beat representation considerably and creates a quantized temporal grid of approximately 23 ms/frame. The network used a validation split of 0.1, reserving the last 10% of the data to be used for validation and generative seeds.

To create the input to the LSTM, the beat onsets vectors are windowed into sequences of 20 frames with the sequences being trained to predict the 21st frame (the window then moves over one time step to predict the 22nd frame until the end of the sequence is reached). This sequence length[6] was chosen with the intuition that the performer would be able to respond to external stimuli at around 0.5 seconds which is the equivalent of around 20 frames.

[6] The sequence length was found to be a very significant parameter. I experimented with different sequence lengths to see what effect it would have on the neural network. This is discussed in the results section.

The amplitudes of the beat vectors were either fixed to a *binary* representation (based on the onset amplitude being greater than some threshold value) or were quantized into ten steps from [0.0, 0.1, … 1.0] to create a *scaled* representation. The target values were converted into a one hot encoding at the output so the network could be configured to predict the probability of these 10 quantized intensity values. Whereas the binary beat vectors might only learn lower-level temporal structure, a scaled representation might uncover the ways in which the beat patterns contain accent-oriented (intensity) relationships (e.g. a strong beat followed by a weak beat). The results section comments on some of these characteristics.

Network Architecture

The basic neural network architecture shown in Figure 5 was created using the Keras[7] library for python. For the LSTM-RNN, the first and second LSTM layers contained 50 and 100 units respectively and the dense output uses a sigmoid layer for the binary vector case and a softmax layer of dimension 10 for the scaled case. It uses a dropout with a probability of 20 in between each hidden layer. In short, this is a single value classification problem with either 2 classes (0,1) or 10 classes depending on the binary or scaled representation.

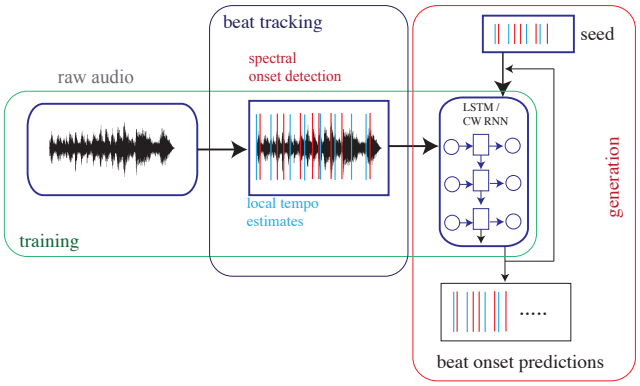

Figure 5: Network Architecture

The CW-RNN consisted of 128 hidden units that were divided into nine equally sized grouped with exponential clock timings (1,2,4,…,256) as parameterized in the original paper's description (J. Koutník et al. 2014). In a CW-RNN, the hidden layer is broken up into separate modules that process inputs at different periodic frequencies. This has the advantage of allowing updates to occur at different clock speeds so that low-frequency modules might impart relevant temporal information to high-frequency modules in a one-directional manner.

Because the fully-connected output is a probability prediction, the error function is categorical cross-entropy or the scaled LSTM and CW-RNN case and a binary cross-entropy for the binary representation LSTM. The network used 'adam' optimization during training. The networks were trained for 200 epochs each which was chosen heuristically based off of the imperative of seeking a balance between generalization of input data without overfitting.

[7] http://github.com/fchollet/keras/

Results and analyses

Because the primary goal of this study was to create a generative model that was capable of producing beat patterns representative of a particular drummer's pulse trajectory, there are several ways in which to evaluate the learned models. First we can look at how well the models were able to predict target samples from the validation set in terms of accuracy.

The binary representation's average prediction accuracy, 60.9%, was slightly better than on-off beats onsets being selected at random. Because the prediction is binary, this means that this model was mostly unable to predict subsequent beats from the test sets. However, the scaled representation showed a prediction accuracy of 73.2 % after 200 epochs. The best accuracy was achieved with the CW-RNN which showed an accuracy of 87.1%.[8]

Figure 6 shows the respective loss over two hundred epochs for the binary LSTM, scaled LSTM and clockwork RNN neural network models for different sequence lengths. The loss was shown to continue to decrease upon further training but for purposes of comparison, the plots for each net were all trained for 200 epochs.

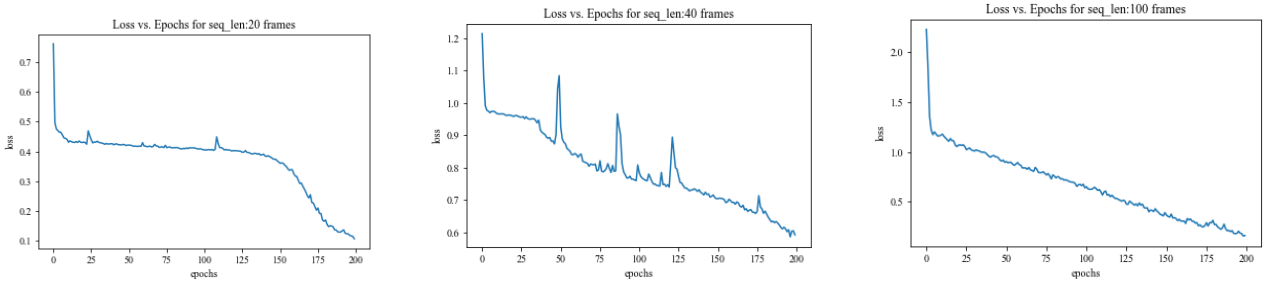

Figure 6: (left to right) Binary LSTM (a), Scaled LSTM (b), and Scaled CW-RNN Loss vs. Epoch

To generate output from the models after training, sequences from the test set were used to *seed* the model to produce an output prediction which were then be fed back into the input. This process is repeated to produce novel sequences of beat activations that the neural network has learned.

The probability density (PDF) plots shown in Figure 7 are helpful in understanding the relative frequency of beat onsets within the length 20 sequences in the training set and their corresponding generated sequences. These plots were produced using a PDF kernel density estimate (kde) to provide a density estimation measure of the test data and the resultant generated sequences (test sequences and generated sequences were assumed to

[8] Because this study was less concerned with predicting actual beat onsets given a prior beat context, this accuracy metric is less meaningful than evaluating if the model has learned a stylistically representation of a drummer's beat pattern.

be non-parametric[9]). It provides an indication of the relative beat onset density of the data within a window of sequence length 20.

In general, the onset density in the generated sequences were sparser than the sequences in the test set. It's evident from Figure 7 that the binary beat onsets in the test set were relatively evenly-distributed with slightly more emphasis on non-beat (0) values which implies that there were more 'rests' than beat events. This characteristic is more pronounced in the LSTM and CW-RNN density plots where there clearly is a dearth of high-valued beat onsets. This is an important feature of the data representation insofar as it provides an indication of beat onset *sparsity* as it applies to both training and tests sets. Similarly, we can look at individual sequences' density distributions to gauge how accurate their predicted generative sequences were with respect to the input.

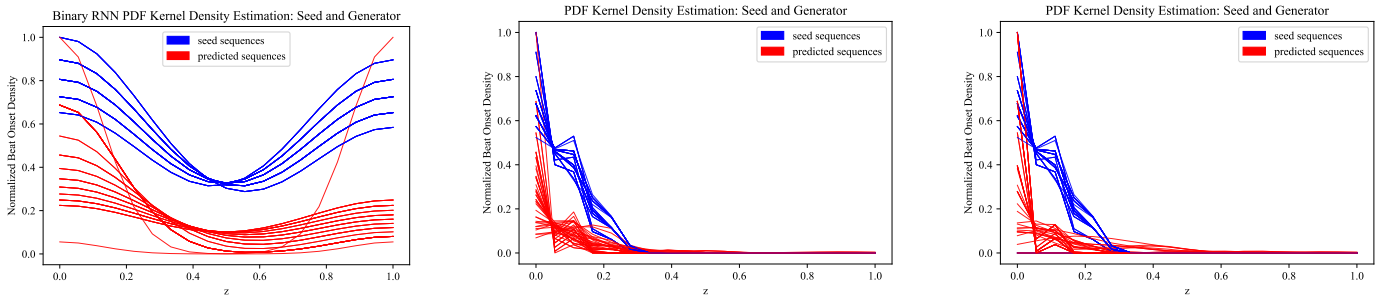

Figure 7: Density Estimate Measure: Kernel Density Estimation of PDF - Cumulative Seed and Generative Sequences for Binary LSTM, Scaled LSTM, and Scaled CW-RNN

Two example generated sequences and their density functions are plotted in Figure 8 and Figure 9. In Figure 7, the binary predicted sequence is more sparsely populated than the seeding sequence. This is reflected in the density plot where the predicted density contour (shown in red) is skewed toward 0.0 which means that it contains a sparser distribution of onsets relative to the seeded sequence. Similarly, Figure 8 shows a generated sequence that contains nearly the same density of beat onsets as the sequence that seeded it which is reflected in the nearly overlapping density estimation.

[9] I would be interested in applying statistical methods to the beat onset data set. If the beat onset data was found to be associated with a parametric family of distributions, then one could assume that density estimation could be determined by finding estimates of the mean and variance in the data.

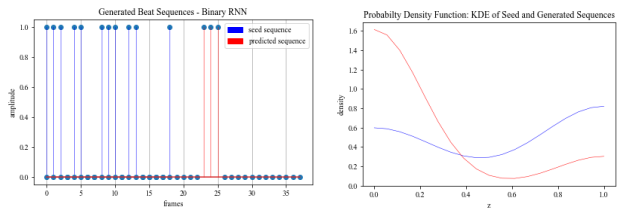

Figure 8: Density Estimation of 'Dissimilar' Seed and Generated Sequences.

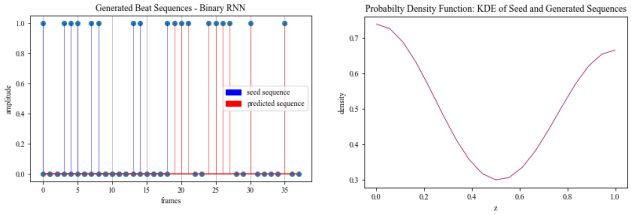

Figure 9: Density Estimation of 'Similar' Seed and Gen. Sequences

Discussion

All of the neural nets employed in this study were able to learn information pertaining to the onsets' density per frame window while the scaled LSTM and CW-RNN were shown to be somewhat capable of learning mid-level pulse representation with intensity contours reflective of the test set. The CW-RNN was shown to perform significantly better in terms of producing novel output while still retaining *stronger* beats that reflect the global tempo estimate onsets. The binary beat representation was mostly only able to encode the beat density and was generally unable to produce novel sequences containing pulse.

One of the main determinants of the generative output was the length of the sequence window that parses the training examples. This is the number of frames the network trains on to predict the subsequent beat onset. Determining the optimal sequence length was mostly done heuristically until a sequence length of 20 was chosen as a benchmark. 20 frames also correspond to approximately 0.5 seconds which seemed like a reasonable amount of time for drummer to respond to acoustic stimuli in the course of a performance.

Most of the generated sequences in the test set were able to produce novel beat sequences for about 20 to 30 time steps. This is probably due to the fact that they were trained on sequences of roughly the same length. By modulating the sequence prediction window to 50 frames, the networks were generally unable to generate output that contained both periodic and novel patterns. Conversely, decreasing the sequence window to around 5 frames the model was able to pick up on more locally periodic patterns in the input but was unable to provide longer term generative output that did not settle on some predetermined repetition.

The outcomes of the generated sequences were very sensitive to the specific seeding data that produced it. The two main types of generative trends produced from seeding the models were indefinite repetition upon subsequent time steps, settling on 0, or production of quasi-periodic beat patterns. However, many of the seeded inputs from the test set were able to induce a periodic rhythm at the output that was reflective of a pulse from the input sequence. For instance, the clockwork RNN generated sequence shown in Figure 10 shows a predicted sequence that contains repeated pattern that bears a similar temporal grouping from the seeded sequence shown in blue. One way to analyze this behavior is to suggest that the network has learned first-order information pertaining to the seed sequence's local temporal pattern and a second-order grouping derived from the first that pertains to that pattern's global temporal pattern. Figure 11 shows a scaled LSTM network that clearly was able to learn the global tempo estimated onsets from the test set. The two large onsets offset by about 16 frames in the seed sequence is repeated three times in the generated sequence.

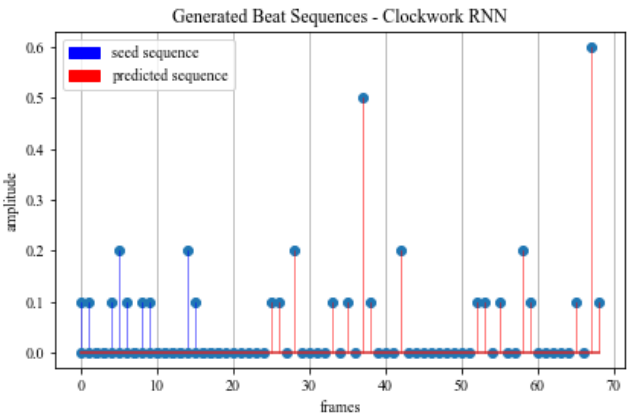

Figure 10: Generated Beat Sequence with locally Periodic Information

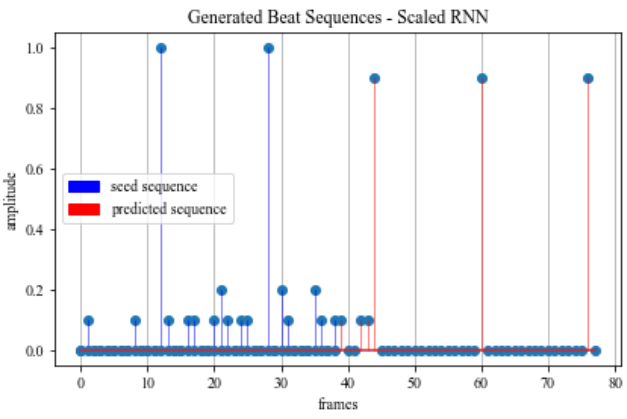

Figure 11: Generated Beat Sequence

Upon increasing the sequence generation length to around 300 frames (which corresponds to roughly 7 seconds) many of the sequences were able to produce periodic yet non-isochronous temporal patterns that seemed to reflect input sequence temporal and intensity relationships. Such an example is show in Figure 12 and 13 where the generated sequence was extended to 300 frames. Figure 9 shows how the network has learned to generate a periodic pulse at about the same frequency as the 1.0 values in the seed sequence. The higher frequency 0.1 values in the seed sequence are also present in the generated sequence and produce novel patterns of activations that are non-isochronous.

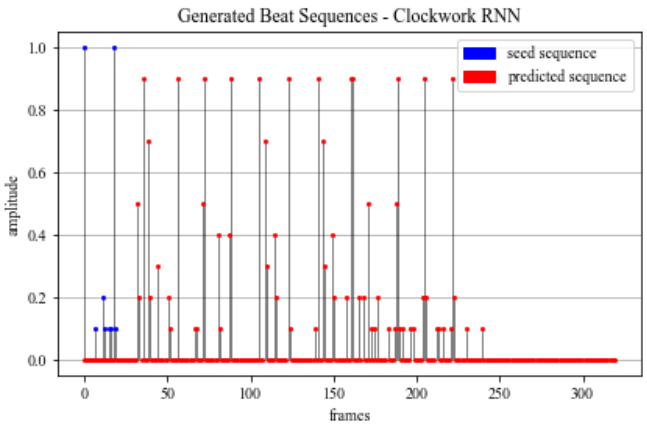

Figure 12: Generated Beat Onsets showing Periodic Pulse Onsets

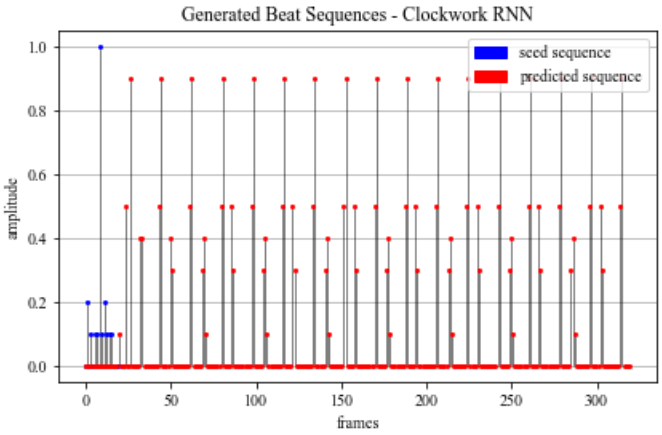

Figure 13. Generated Beat Onsets showing Temporal Hierarchy

The main limitations of this approach is that the neural network is dependent on the pulse being properly represented (as a combination of spectral envelopes and global tempo estimates) as input to the neural net. In this way, the network is only as good as the data it trains on; if this representation of pulse is ill-suited to the task at hand, the network will be unable to learn any patterns of pulse structure. The models proposed in this

study were notionally inspired by the word generation models of LSTM networks (namely their letter-word-sentence context dependence). In this study however time is treated as somewhat analogously to letter space in the LSTM word generation models. One reason the word generation networks can learn to produce (semi) coherent sentences is because of these relational and contextual patterns between letters and words (e.g. certain vowels tend to follow certain consonants in certain orders). The beat onset task described in this paper does not share this same level of consistency between onset values and temporality. In the scaled LSTM-RNN, different onset values (as one hot vectors) would correspond to different letters in the word generation model but these onset values are less likely to contain the same structural relationship within letters via a given word in a given sentence.

Future Work

Another beat representation to feed into the networks would be to create onset vectors that encode the *time differences* of the spectral onsets from the global tempo estimate onsets. In this way, the network would learn deviations from a pulse representation that is pre-coded into the periodicity inherent in the global tempo. The network wouldn't have to learn pulse induction from scratch but could focus on how rhythmic events tend to deviate from expected beats. This schema might possess a capacity to learn stylistic rhythmic 'feels' that are associated with certain types of music (e.g. 'swing' in jazz music, 'deep pocket' in funk).

Nevertheless, as a generative model the neural networks were able to pick up on structural features of the beat onsets in the training set and generate them with some amount of variability in the output. As a future application, this neural network might be pre-trained on much more extensive set of training examples, and then used in real-time with a performer to incorporate rhythmic aspects of their performance into the network. Similarly, a performer might be able to access specific parameters of the network in real-time so as to induce certain output patterns. Additionally, in training the network it would be advantageous to be able to separate individual drums from the acoustic texture itself to form beat activation patterns per drum in the drum ensemble. In this way, the network might be able to learn patterns of activation between percussion instruments in the set as opposed to relying on mid-level, one-dimensional representation of pulse.

References

A. Holzapfel, M. E.P. Davies, J.R. Zapata, J.L. Oliveira, and F. Gouyon, "Selective sampling for beat tracking evaluation," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 9, pp. 2539–2548, 2012.

Bello, J.P. Daudet, L. Abdallah, A. Duxbury, C. Davies, C. & Sandler, M. "A tutorial on onset detection in music signals," *Speech and Audio Processing, IEEE Transactions on*, vol. 13, no. 5, pp. 1035–1047, 2005.

D.P.W. Ellis, "Beat Tracking by Dynamic Programming," *Journal of New Music Research*, Vol. 36(1), 51–60, 2007.

Davies, M.E.P. Degara, N, & Plumbley, M. "Evaluation methods for musical audio beat tracking algorithms," 2009.

Fitch, T.W. "Rhythmic cognition in humans and animals: distinguishing meter and pulse perception". Front. Syst. Neurosci., 31 October 2013 | https://doi.org/10.3389/fnsys.2013.00068

Graves, M. & Pullen, D. (1967). *nommo* [compact disc]. New Haven, NJ: SRP Records.

Hochreiter,S. Schmidhuber, J. “Long Short-Term Memory,”*Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

J. Koutník, K. Greff, F. Gomez, & J. Schmidhuber. (2014) “A Clockwork RNN.” http://arxiv.org/abs/1402.3511

Jones, M. R. (1976). Time, Our Lost Dimension: Toward a New Theory of Perception, Attention, and Memory. *Psychological Review*, 83. 323-355.

Karpathy, A. “The Unreasonable Effectiveness of Recurrent Neural Networks”. (2015, May 21). Andrej Karpathy Blog [Blog post]. Retrieved from http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Kirschner, S. & Tomasello, M. (2009). Joint drumming: Social context facilitates synchronization in preschool children. *J Exp Child Psychol 102*, 299–314

Lambert, A., Weyde, T. & Armstrong, N. (2014). *Studying the Effect of Metre Perception on Rhythm and Melody Modelling with LSTMs*. Paper presented at the 3rd International Workshop on Musical Metacreation, held at the 10th Artificial Intelligence and Interactive Digital Entertainment Conference, 03-10-2014 - 07-10-2014, Raleigh, USA.

Large, E. W., Herrera J. A. and Velasco M. J. (2015). Neural networks for beat perception in musical rhythm. *Frontiers in Systems Neuroscience*. *9* (159). doi: 10.3389/fnsys.2015.00159

Large, E.W, Almonte, F. V, & Velasco, M. J. 2010. A canonical model for gradient frequency neural networks. *Physica D*, 239. 905-911.

Lerdahl, F. & Jackendoff, R. *A Generative Theory of Tonal Music*, Cambridge, Mass. MIT Press. 1983.

Peeters, G. “Time variable tempo detection and beat marking,” in *Proc. ICMC*, 2005.

Sidor, S and Jaques, M. Google Magenta (2016) GitHub Repository. https://github.com/tensorflow/magenta/

Ulyanov, D. Lebedev, V. “Audio Texture Synthesis and Style Transfer” [Blog Post]. Retrieved from https://dmitryulyanov.github.io/audio-texture-synthesis-and-style-transfer/