Studying the Effect of Metre Perception on Rhythm and Melody Modelling with LSTMs

Andrew Lambert and **Tillman Weyde** and **Newton Armstrong**
City University London

Abstract

In this paper we take a connectionist machine learning approach to the problem of metre perception and melody learning in musical signals. We present a multi-layered network consisting of a nonlinear oscillator network and a recurrent neural network. The oscillator network acts as an entrained resonant filter to the musical signal. It 'perceives' metre by resonating nonlinearly to the inherent periodicities within the signal, creating a hierarchy of strong and weak periods. The neural network learns the long-term temporal structures present in this signal. We show that this network outperforms our previous approach of a single layer recurrent neural network in a melody and rhythm prediction task.

We hypothesise that our system is enabled to make use of the relatively long temporal resonance in the oscillator network output, and therefore model more coherent long-term structures. A system such as this could be used in a multitude of analytic and generative scenarios, including live performance applications.

1 Introduction

Beat induction allows us to tap along to the beat of music, perceiving its pulse. This perceived pulse can be present in the stimulus, but it is often only implied by the musical events. Furthermore, performed music is rarely periodic and is subject to the performers' expressive timing. This makes beat induction difficult to model computationally.

Finding the pulse within a musical signal is a step towards achieving other music perception tasks, such as metre perception. Metre refers to the multi-layered divisions of time present in music, of which the referent layer is the pulse. Other layers in music divide the pulse into the smallest subdivisions of time, and extend it towards larger measures, phrases, periods, and even higher order forms. Thus, a single 'beat' can occur at one or more metrical levels, whereas the 'pulse' is the series of beats on the referent layer only. A beat on multiple metrical levels is perceived to be 'stronger' than other beats, creating a beat hierarchy, or metrical structure (**?**). The individual components of music, the rhythmic events in time, lead to the formation of new macroscopic spatial, temporal and functional structures in metre. In performance, these structures vary and repeat with time in their own patterns.

The process through which humans achieve beat induction is known as entrainment. Entrainment is the coordination of temporally structured events through interaction where two or more periodic signals are coupled in a stable relationship. Many relationships are possible in entrained signals, exact synchronisation is considered to be a special case of entrainment. Ethnomusicologists are increasingly becoming aware of the importance of entrainment processes as an approach to understanding music making and music perception as a culturally interactive process (**?**).

Much prior work on pulse and metre perception has been concerned with abstract temporal information, such as crafted pulses in time (**?**; **?**; **?**; **?**). However, metre perception and preference develops through cultural learning and is determined by a multitude of musical signposts, including the melody and the tempo of the pulse (**?**; **?**).

This project aims to support melody and rhythm modelling in a recurrent neural network by using an oscillator layer for metre perception. We are evaluating the network in different configurations and with different note representations on a melody prediction task.

2 Models

Our network consists of two connected networks. The first is a Gradient Frequency Neural Network (GFNN), a nonlinear oscillator network (**?**). It acts as an entrained resonant filter to the musical signal and serves as a metre perception layer. The second is a Long Short-Term Memory network (LSTM) (**?**), a recurrent neural network, which is able to learn the kind of long-term temporal structures required in music signal prediction (**?**).

Metre Perception Layer

Oscillators have been used for beat induction in machines for over twenty years. Certain oscillator models lend themselves well to beat induction tasks due to their stable limit cycle and their entrainment properties (**?**). By using oscillators to perceive beats, we have the ability to model beat induction as an emergent dynamical process, which changes over time as the signal itself evolves. Gasser et al.'s SONOR system, for instance, adds Hebbian learning to networks of

adaptive oscillators, which can then learn to produce a metrical pattern (**?**).

More recently, the phenomenon of nonlinear resonance has been applied to metre perception and categorisation tasks. Large et al. (**?**) have introduced the Gradient Frequency Neural Network (GFNN), which is a network of oscillators whose natural frequencies are distributed across a spectrum. When a GFNN is stimulated by a signal, the oscillators resonate nonlinearly, producing larger amplitude responses at certain frequencies along the spectrum. Nonlinear resonance can account for pattern completion, the perception of the missing fundamental, tonal relationships and the perception of metre (**?**).

When the frequencies in a GFNN are distributed within a rhythmic range, resonances occur at integer ratios to the pulse. These resonances can be interpreted as a hierarchical metrical structure. Rhythmic studies with GFNNs include rhythm categorisation (**?**), beat induction in syncopated rhythms (**?**) and polyrhythms (**?**).

Temporal Structure Layer

There have been many connectionist approaches to musical tasks, e.g. (**?**; **?**; **?**; **?**; **?**). Whilst recurrent neural networks are good at learning temporal patterns, they often lack global coherence due to the lack of long-term memory. Long Short-Term Memory (LSTM) networks were designed to overcome this problem. A simplified diagram of an LSTM memory block can be seen in Figure **??**. A self-connected node known as the Constant Error Carousel (CEC) ensures constant error flow back through time. The input and output gates control how information flows into and out of the CEC, and the forget gate controls when the CEC is reset. The input, output and forget gates are connected via 'peepholes'. For a full specification of the LSTM model we refer to (**?**).

LSTMs have already had some success in music applications. Eck and Schmidhuber (**?**) trained LSTMs which were able to improvise chord progressions in the blues and more recently Coca et al. (**?**) used LSTMs to generate melodies that fit within user specified parameters.

3 Experiments

Our experiments operate on monophonic symbolic music data. We have used a corpus of 100 German folk songs from the Essen Folksong Collection (**?**).

We conducted all experiments in two steps, implementing the GFNN in MATLAB[1] using the standard differential equation solvers, and the LSTM in Python using the PyBrain[2] library.

GFNN

The GFNN consisted of 128 Hopf oscillators defined by the following differential equation:

$$\frac{dz}{dt} = z(\alpha + i\omega + \frac{\beta\varepsilon|z|^4}{1-\varepsilon|z|^2}) + \frac{x}{1-\sqrt{\varepsilon}x}.\frac{1}{1-\sqrt{\varepsilon}\bar{z}} \quad (1)$$

[1] http://www.mathworks.co.uk/

[2] http://pybrain.org/

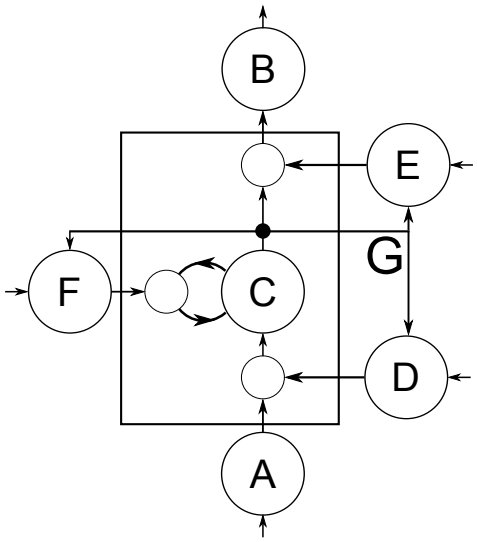

Figure 1: A single LSTM memory block showing (A) input, (B) output, (C) CEC, (D) input gate, (E) output gate, (F) forget gate and (G) peephole connections.

where z is the complex valued output, $\bar{z}$ is its complex conjugate, ω is the driving frequency in radians per second, α is a damping parameter, β is an amplitude compressing parameter, ε is a scaling parameter and x is a time-varying stimulus. This oscillator is complex valued, oscillates spontaneously according to its parameters, and entrains to and resonates with an external stimulus. For all experiments, parameter values were fixed as follows: $\alpha = -0.1, \beta = -0.1, \varepsilon = 0.5$.

This gives a sinusoid-like oscillation whose amplitude is gradually dampened over time (see Figure **??**). The gradual dampening of the amplitude allows the oscillator to maintain a long temporal memory of previous stimulation.

The oscillator frequencies in the network were logarithmically distributed from 0.25Hz to 16Hz. The GFNN was stimulated by rhythmic time-series data in the form of a decay envelope on note onsets, synthesised from the symbolic data. All sequences in the corpus were synthesised at a tempo of 120bpm (2Hz), meaning that our metrical periodicities the GFNN ranged from a demisemiquaver (32nd note) to a breve (double whole note).

Performing a Fourier transform on the GFNN output reveals that there is energy at many frequencies in the spectrum, including the pulse (Figure **??**). Often this energy is located at integer ratios to the pulse, implying a perception of the metrical structure.

LSTM

All experiments used the standard LSTM model with peephole connections enabled and the number of hidden LSTM blocks fixed at 10, with full recurrent connections. The number of blocks was chosen empirically as it provided reasonable prediction accuracy with plenty of potential for improvement, whilst minimising the computational complexity of the LSTM. Training was done by backpropagation through time (**?**) using RProp$^-$ (**?**). During training we used

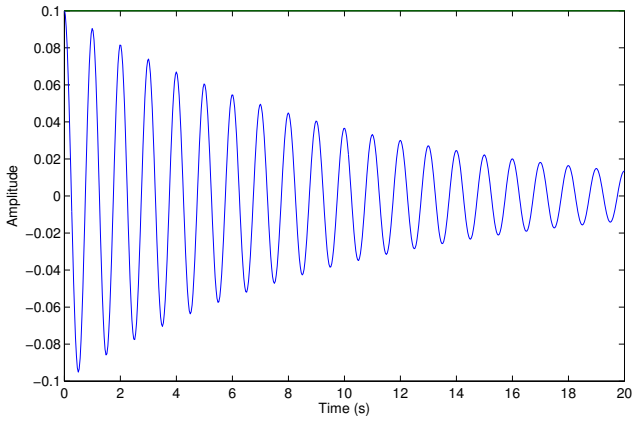

Figure 2: A Hopf oscillator with the following parameters, $\omega = 2\pi, \alpha = -0.1, \beta = -0.1, \varepsilon = 0.5$. The amplitude has decayed by half in approximately 6.5 seconds.

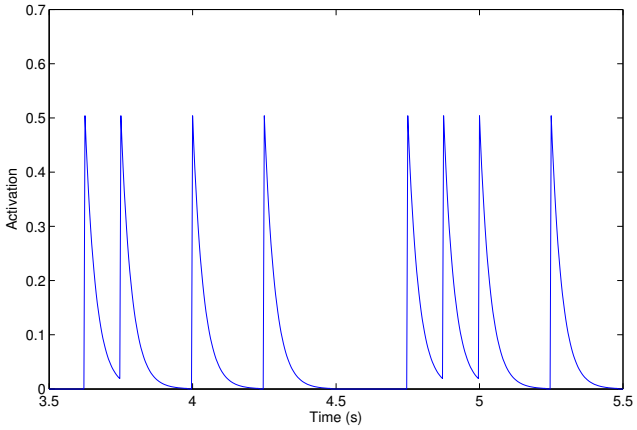

Figure 3: Example note onset time-series data.

k-fold cross-validation (**?**). In k-fold cross validation, the dataset is divided into k equal parts, or 'folds'. A single fold is retained as the test data for testing the model, and the remaining k - 1 folds are used as training data. The cross-validation process is then repeated k times, with each of the k folds used exactly once as the test data. For our experiments k was fixed at 4, and a maximum of 2500 training epochs was set per fold, but never reached. We also evaluated on the training data and found a mean percentage increase across all metrics of no more than 4.4%, indicating a good generalisation without over-fitting.

Experiment 1: Pitch Prediction

Our first experiment was designed to investigate the effect of adding the metre data from the GFNN to a pitch prediction task. We created three LSTMs, all of which were tasked with predicting pitch in the form of time-series data.

We abstracted the absolute pitch values to their relative scale degrees to keep the model simple in these initial experiments. Accidentals were encoded by adding or subtracting 0.5 from the scale degree and rests were encoded as 0 values. We first inserted scale degree numbers, their onsets and offsets into the data stream and then re-sampled the data using the zero-order hold method, such that one sample corresponds to a demisemiquaver. An example data stream can be seen in Figure **??**.

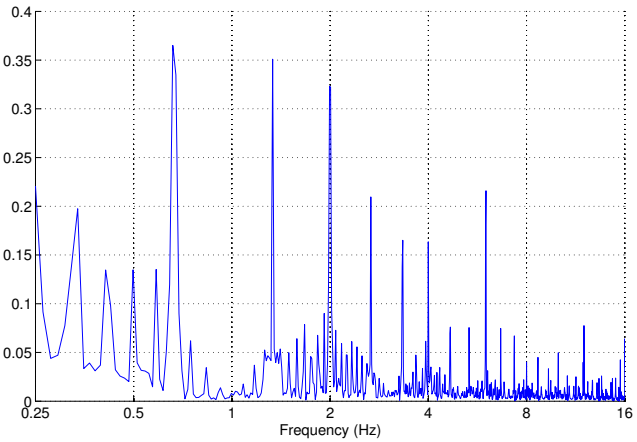

Figure 4: An example magnitude spectrum of a summed GFNN output.

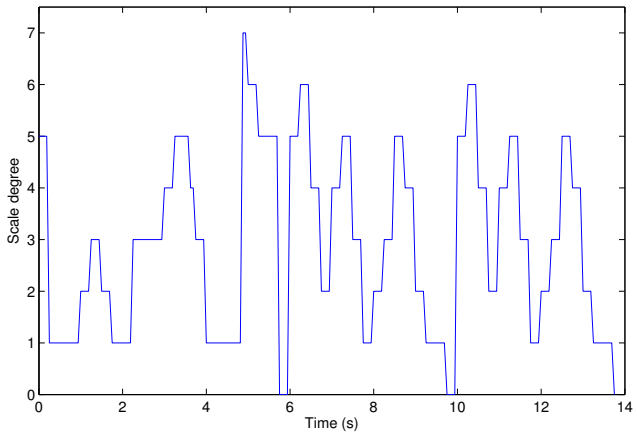

Figure 5: Example scale degree time-series data.

The first network (LSTM1a) was designed as a baseline to measure the impact of the GFNN. It took no input from the GFNN, and so consisted of single input containing the time-series scale degree data from the corpus. We constructed two further networks, one with 128 inputs for each oscillator in the GFNN (LSTM1b), and one with 8 inputs consisting of a filtered GFNN output (LSTM1c). LSTM1a, LSTM1b and LSTM1c are illustrated in Figures **??** and **??**.

As shown in Figure **??**, a GFNN signal has relatively few resonant peaks of energy, therefore many oscillators would be irrelevant to the LSTM. Thus, we hypothesised that the filtered output would make learning easier. The input to LSTM1b was filtered to retain the strongest resonant oscillations in the GFNN. The signal was averaged over the corpus

and the oscillators with the greatest amplitude response over the final 25% of the piece were found. We ensured a spread of frequencies by ignoring frequencies if another near frequency was already included. The selected oscillators were then used for all sequences.

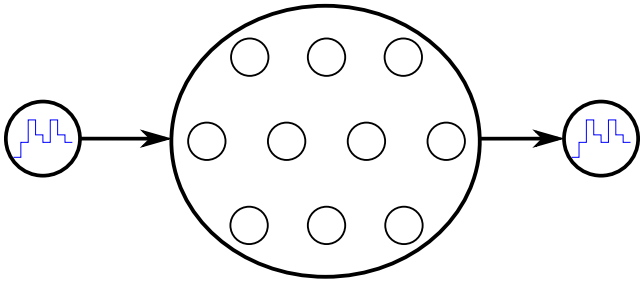

Figure 6: Network diagram for LSTM1a, there is no input from the GFNN.

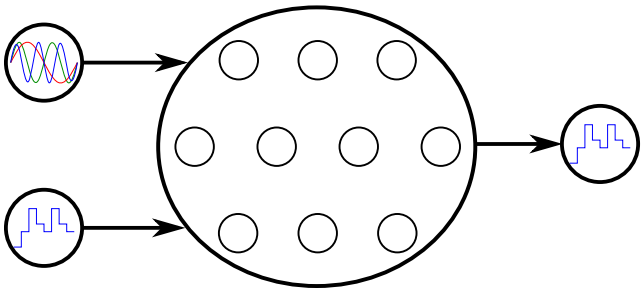

Figure 7: Network diagram for LSTM1b and LSTM1c. LSTM1b had full connections of 128 oscillations from the GFNN, LSTMc had filtered connections of 8 oscillations from the GFNN.

Results Networks were evaluated by activating each of them with the sequences in the corpus (ground truth). We activated the networks with the ground truth throughout the sequence, and for the last 75% of inputs the network output was compared to the target data.

The results have been evaluated using several metrics. Firstly we can see the mean squared error (MSE), which is what the networks were optimised for during training. This provides a view of how close the output was to the target, with a lower number meaning higher accuracy. The next three results refer to the position of pitch changes using standard precision, recall and F-measure, where higher is better. Finally we have a pitch only metric named "Sequence". This has been calculated as a proportion of samples where the output scale degree matches the target value, where again higher is better. Output values were rounded to the nearest half before this comparison was made.

Pitch and rhythm are highly related, but have been singled out here to more fully understand the GFNNs effect on the network. The MSE and sequence metrics represent timing and value, whereas the onset metrics of precision, recall and F-measure represent timing only.

Table **??** shows the results tested against the validation data. The values shown the mean values calculated over the 4 folds in the cross-validation.

We can see from the results that the filtered input from the GFNN (LSTM1c) performed the best at predicting pitch and rhythm. However, there is a striking imbalance between the precision and recall scores for all networks, suggesting a chaotic output from the LSTMs with too many events being triggered. This lead to results that were not impressive overall, with pitch prediction improved, but rhythmic prediction performing poorly.

Experiment 2: Onset Prediction

With our next experiment we wanted to investigate if the GFNN did indeed contain useful rhythmic information for the LSTM to learn. We designed a simpler task where the LSTM had to predict the onset pattern used to stimulate the GFNN from the GFNN data only.

We created two networks for this task: LSTM2a and LSTM2b. LSTM2a had a full GFNN input, and LSTM2b had the same filtered input from the previous experiment. Both networks had one output and were trained to reproduce the GFNN stimulus seen in Figure **??**. A network diagram can be seen in Figure **??**.

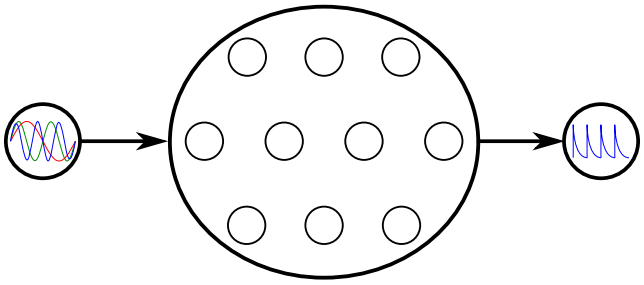

Figure 8: Network diagram for LSTM2a and LSTM2b. LSTM2a had full connections of 128 oscillations from the GFNN, LSTM2b had filtered connections of 8 oscillations from the GFNN.

Results Table **??** shows the results when the networks are tested against the validation data.

All networks were evaluated as in experiment 1, except we no longer have a sequence metric but include the Pearson product-moment correlation coefficient (PCC). This gives a relative rather than absolute measure of how close the target and output signals match, with higher values representing closer matches. LSTM2a performed the best at this task in all metrics, however it is clear from the results that both LSTM2a and LSTM2b perform the tasks well.

The fact that LSTM2a outperformed LSTM2b shows that the LSTM network was able to train itself to ignore the noise produced by the GFNN. It also shows that the GFNN data contains useful information in the weaker resonances that the filtering process removed. Our filtering process may have been too aggressive in this respect. However, having noted this, LSTM2b did not completely fail at the task, therefore a

Network	MSE	Precision	Recall	F-measure	Sequence
LSTM1a	0.75836	0.12154	0.34366	0.17425	0.67107
LSTM1b	0.74115	0.18644	**0.78908**	0.29838	0.47756
LSTM1c	**0.68866**	**0.22852**	0.70196	**0.34137**	**0.69459**

Table 1: Results of the pitch only experiment.

Network	MSE	PCC	Precision	Recall	F-measure
LSTM2a	**0.01277**	**0.79400**	**0.82362**	**0.82769**	**0.82265**
LSTM2b	0.01380	0.77395	0.79411	0.81157	0.79564

Table 2: Results of the onset only experiment.

more permissive filtering technique may still produce better results than even LSTM2a.

Experiment 3: Onset and Pitch Prediction

Experiment 2 has shown us that the GFNN output can be used to reconstruct onsets. Experiment 3 was designed to investigate if tasking the network to directly predict the onsets could aid the prediction of pitch data. We therefore combined experiments 1 and 2, resulting in LSTMs with two outputs: one for pitch and one for onsets. We constructed

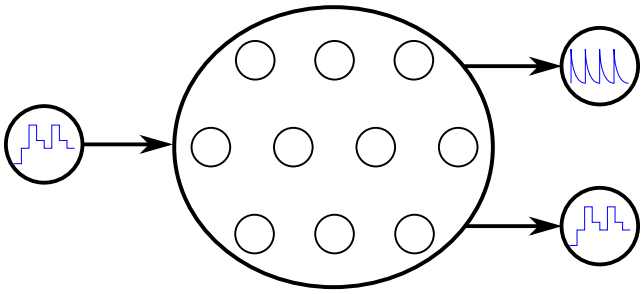

Figure 9: Network diagram for LSTM3a. There is no input from the GFNN.

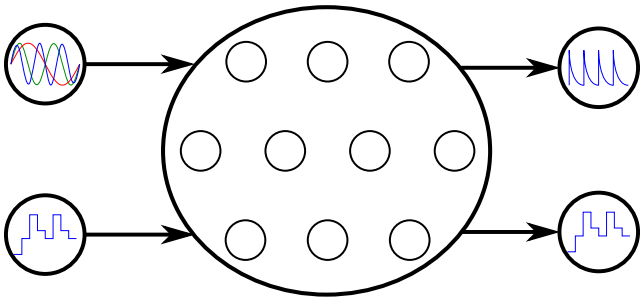

Figure 10: Network diagram for LSTM3b and LSTM3c. LSTM3b had full connections of 128 oscillations from the GFNN, LSTM3c had filtered connections of 8 oscillations from the GFNN.

three LSTMs to conduct this experiment, following the same pattern as experiment 1: no GFNN input, full GFNN input, and filtered input. Network diagrams can be seen in Figures **??** and **??**.

Results All networks were evaluated in the same way as experiments 1 and 2. The MSE metric was calculated for both outputs, PCC, precision, recall and F-measure were only calculated for the onset pattern output, and sequence was calculated only for the pitch output. Table **??** shows the results against the validation data.

We can see from the results that LSTM3c was the best overall network. Whilst LSTM3a did score a better MSE, it scored very poorly on the onset prediction task. This shows that MSE may not be the best optimisation target during training.

In experiment 1, all LSTMs suffered from poor precision scores. Judging by the onset scores, the GFNN input in LSTM3b and LSTM3c leads to great improvement on this. However, an evaluation of the pitch changes comparable with experiment 1 remains to be done.

In experiment 2, the fully connected LSTM2a outperformed the filtered LSTM2b on onset prediction, whereas in this experiment the reverse is true. This could be due to the increased complexity of the problem. The introduction of pitch modelling may have prevented the LSTM learning from the GFNN data effectively, so that the filtering process was beneficial. We can take what we have learned from experiment 1 and hypothesise that an improved filtering method may further improve results. Increasing the number of hidden LSTM blocks may also improve results for both LSTM3b and LSTM3c.

The sequence scores for all networks are somewhat worse in this experiment when compared to experiment 1. However, the improved onset prediction indicates that LSTM3b and LSTM3c are more stable. More work is needed to investigate the behaviour of the pitch prediction to sequence accuracy and stability.

LSTM3c outperformed LSTM3a on the pitch prediction task, whilst also predicting stable onset patterns. This provides evidence that melody models can be improved by modelling metre.

4 Conclusion

We have presented a multi-layered network consisting of a metre perception layer (GFNN), and a temporal prediction

Network	MSE	PCC	Precision	Recall	F-measure	Sequence
LSTM3a	**7.26251**	0.23253	0.35655	0.06368	0.10233	0.64459
LSTM3b	7.34243	0.58499	**0.71622**	0.60717	0.65110	0.58371
LSTM3c	7.32129	**0.62905**	0.70480	**0.76750**	**0.72589**	**0.65755**

Table 3: Results of the pitch and onset experiment.

layer (LSTM). The GFNN output, with its strong and weak nonlinear resonances at frequencies related to the pulse, can be interpreted as a perception of metre. Our results show that providing this data from the GFNN helped to improve melody prediction with an LSTM. We hypothesise that this is due to the LSTM being able to make use of the relatively long temporal resonance in the GFNN output, and therefore model more coherent long-term structures.

In all cases GFNNs improved the performance of pitch and onset prediction, Given the improvements to the onset prediction, modelling pitch and onsets can be seen to be the best overall approach. Additionally, the best results were achieved by filtering the GFNN output. However, experiment 2 shows us that there is important information in the full GFNN signal which is lost through the filtering method adopted here. In addition, this filtering method may not be a good solution when dealing with varying tempos or expressive timing, as it introduces an assumption of a metrically homogeneous corpus. Thus, two tasks for future work are to develop filtering that improves performance and supports tempo variation as well as exploring representations and learning methods that combine stable onset prediction with sequence accuracy.

Both Eck and Schmidhuber's (**?**) and Coca et al.'s (**?**) LSTMs either operate on note-by-note data, or quantised time-series data. By inputting metrical data, our system can be extended to work with real time data, as opposed to the metrically quantised data we are using here. We feel these initial experiments give some indication that better melody models can be created by modelling metrical structures.

By using an oscillator network to track the metrical structure of a performance data, we can move towards real-time processing of audio signals and close the loop in the GFNN-LSTM, creating an expressive, metrically aware, generative real-time model.

5 Acknowledgements

Andrew Lambert is supported by a PhD studentship from City University London.