Adaptive Drum Machine Microtiming with Transfer Learning and RNNs

Grigore Burloiu
CINETic UNATC
grigore.burloiu@unatc.ro

ABSTRACT

We introduce `rolypoly~`, the first drum machine for live performance that adapts its microtiming in relation to a human musician. We leverage state-of-the-art work in expressive performance modelling with recurrent nets, towards real-time application on the micro scale. Our models are pretrained on the Groove MIDI Dataset from Magenta, and then fine-tuned iteratively over several duet performances of a new piece. We propose a method for defining training targets based on previous performances, rather than a prior ground truth. The agent is able to adapt to human timing nuances, and can achieve effects such as morphing a rhythm from straight to swing.

1. INTRODUCTION

We are interested in the coupling between drum machine and human player, with a view to extending their mutual dynamics computationally [1]. We centre on the microtiming scale as a locus of expressive music interaction [2].

In studies on tempo and time-shift representation, [3] posits that global tempo curves alone cannot account for the alterations observed in performances of the same material at different speeds. Nevertheless, score-driven automatic accompaniment has traditionally worked by computing such a curve to drive the warping of a backing track [4, 5]. Increasingly however, attention is also being paid to the *interpretation* of real-time accompaniment on the micro scale [6, 7]. By leveraging advances in natural language processing, recurrent neural network (RNN)-based machine learning architectures have been shown to produce state-of-the-art expressive outputs [8–10].

We delineate a set of design principles for a new musical agent. `rolypoly~` is a score-driven drum machine with the following characteristics: *lightweight* (inference must be fast enough to run in real-time on an average, accessible computer), *audio-interactive* (must not only adapt to incoming sound from human musicians, but also "listen" to its own output, coupled with the former), *progressive/scalable* (must not rely on existing duet/ensemble ground truth; rather, it builds up a performance corpus, accruing learning along the way), and *score-agnostic* (does not require symbolic specification of the parts played by other musicians).

2. AGENT ARCHITECTURE

The piece to be interpreted is represented as a sequence of feature vector rows, encoding each drum hit and its context (tempo, time signature, beat phase) at timestep t, and the drum-target onset distance $\widehat{diff}_{t-1}$ (since this distance can only be measured in hindsight). The output is a drum microtiming offset, y_t, with its corresponding estimation $\hat{y}_t$ determining when in relation to the absolute notated time the t-th drum hit will actually be triggered.

Since we do not rely on an existing drum-human duet ground truth (no such dataset suitable for deep learning exists), we define y after and as a function of a performance, as follows.

We define a variable d_t at timestep t as the cumulated realised offsets of score-to-drum and (variance-adjusted) drum-to-target:

$$d_t = \hat{y}_t + A \cdot \frac{\widehat{diff}_t}{\sigma_{\widehat{diff}} / \sigma_{\hat{y}}} \,, \tag{1}$$

We then use d_t to determine the ground truth drum offsets for training the next iteration of the model, by subtracting its mean (to keep outputs centred around zero) and again applying deviation normalisation:

$$y_t = B \cdot \frac{d_t - \mu_d}{\sigma_d / \sigma_{\hat{y}}} \,. \tag{2}$$

A and B above are hyperparameters, controlling the weighting of the target offsets and the cumulated offsets, respectively. Their default setting is 1, allowing the timing output to adapt gradually, without major fluctuations. For rhythm morphing (see Section 3) they can be set to cancel out the effect of variance scaling.

We propose two RNN-derived architectures that predict each following drum timing based on the input features received in real time. Both are defined and trained using the PyTorch library, and communicate via OSC to Max, which handles the audio playback and analysis.

The first model is built on a 2-layer unidirectional LSTM network [11] with 256 hidden units fed into *tanh* nonlinear activations. The result is passed through a linear layer with a single output, $\hat{y}$.

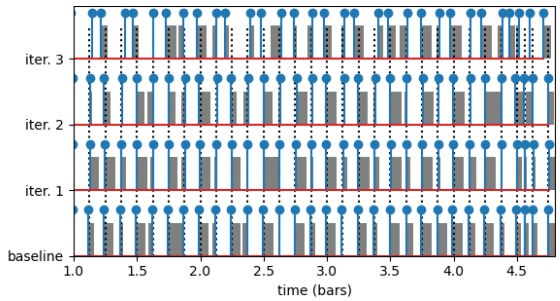

Figure 1. Rhythm morphing over three training iterations. Score: dotted lines. $\hat{y}$: blue stems. $\widehat{diff}$: grey bars. Transition from straight (bottom) to swinging (top).

The second model is a simplification of the Seq2Seq [12] architecture described in [9]. In our case the source sequence is the complete pre-performance input dataset (*sans* the unrealised audio-related features), and the target is the performance dataset up to the current timestep. The encoder is a bidirectional LSTM, and the decoder is a 2-layer unidirectional LSTM with 256 hidden units. As with the first model, a *tanh* nonlinearity and final linear layer project the decoder output to a one-dimensional activation, $\hat{y}$.

To pretrain our models we process a drums-only performance dataset, the Groove MIDI Dataset (GMD) from Magenta [9], the largest existing dataset of expressive drumming. We predict residual drum offsets, [1] resulting in audio-agnostic expressive interpreter models, ready to be fine-tuned with subsequent performances.

The source code and trained models [2], notebook [3] and a demo video [4] are available online.

3. EXPERIMENTS

Figure 1 pictures the Seq2Seq model transitioning from a straight 4/4 beat to a "swing" shuffle, where offbeats are pushed slightly later—simply by "swinging" the respective notes on guitar for three iterations. Similarly we were able to morph the pattern `x..x..x.` to three equally-distanced triplets, as seen in the demo video.

The typical `rolypoly~` use case consists in a song being performed multiple times and the model learning incrementally after each take. Heuristically we found the agent is able to limit the drum-target variance, and adapt to structural patterns in the piece.

4. FUTURE WORK

We are exploring improvements to e.g. the target audio representation, via the learning of a latent feature space of descriptors [7]. Presently, we are adding velocity to the model, and porting the inference code into a Max external using LibTorch. All the above are to be approached in the context of building a multi-timescale hierarchical system.

[1] Thus, y measures the distance to the drum hit from its quantised position. While [9, 13] used 16 steps per bar, we chose a quantisation step of 24, to better account for triplets and swing.

[2] See `https://github.com/RVirmoors/rolypoly/tree/master/py`.

[3] See `https://github.com/RVirmoors/rolypoly/blob/master/py/rolypoly.ipynb`.

[4] See `https://youtu.be/UHBIzfc5DCI`.

5. REFERENCES

[1] M. d'Inverno and J. McCormack, "Heroic versus collaborative AI for the arts," 2015.

[2] M. Leman, *The expressive moment: How interaction (with music) shapes human empowerment*. MIT press, 2016.

[3] H. Honing, "Timing is tempo-specific," in *Proceedings of the International Computer Music Conference*, 2005, pp. 359–362.

[4] C. Raphael, "Music plus one and machine learning," in *International Conference on Machine Learning (ICML)*, 2010.

[5] A. Cont, "On the creative use of score following and its impact on research," in *SMC*, 2011.

[6] G. Xia, Y. Wang, R. B. Dannenberg, and G. Gordon, "Spectral learning for expressive interactive ensemble music performance," in *ISMIR*, 2015, pp. 816–822.

[7] A. Maezawa, "Deep linear autoregressive model for interpretable prediction of expressive tempo," *Proc. SMC*, pp. 364–371, 2019.

[8] D. Jeong, T. Kwon, and J. Nam, "Virtuosonet: A hierarchical attention rnn for generating expressive piano performance from music score," in *NeurIPS 2018 Workshop on Machine Learning for Creativity and Design*, 2018.

[9] J. Gillick, A. Roberts, J. Engel, D. Eck, and D. Bamman, "Learning to groove with inverse sequence transformations," *Proceedings of the 36th International Conference on Machine Learning, PMLR*, pp. 2269–2279, 2019.

[10] S. Oore, I. Simon, S. Dieleman, D. Eck, and K. Simonyan, "This time with feeling: Learning expressive musical performance," *Neural Computing and Applications*, vol. 32, no. 4, pp. 955–967, 2020.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[13] D. Makris, M. Kaliakatsos-Papakostas, I. Karydis, and K. L. Kermanidis, "Conditional neural sequence learners for generating drums' rhythms," *Neural Computing and Applications*, vol. 31, no. 6, pp. 1793–1804, 2019.