Deep Models for Ensemble Touch-Screen Improvisation

Charles P. Martin
Department of Informatics
charlepm@ifi.uio.no

Kai Olav Ellefsen
Department of Informatics
kaiolae@ifi.uio.no

Jim Torresen
Department of Informatics
jimtoer@ifi.uio.no

ABSTRACT

For many, the pursuit and enjoyment of musical performance goes hand-in-hand with collaborative creativity, whether in a choir, jazz combo, orchestra, or rock band. However, few musical interfaces use the affordances of computers to create or enhance ensemble musical experiences. One possibility for such a system would be to use an artificial neural network (ANN) to model the way other musicians respond to a single performer. Some forms of music have well-understood rules for interaction; however, this is not the case for free improvisation with new touch-screen instruments where styles of interaction may be discovered in each new performance. This paper describes an ANN model of ensemble interactions trained on a corpus of such ensemble touch-screen improvisations. The results show realistic ensemble interactions and the model has been used to implement a live performance system where a performer is accompanied by the predicted and sonified touch gestures of three virtual players.

CCS CONCEPTS

• **Computing methodologies** → *Neural networks*; • **Applied computing** → *Sound and music computing*;

KEYWORDS

deep learning, RNN, ensemble interaction, touch screen performance, mobile music

ACM Reference Format:
Charles P. Martin, Kai Olav Ellefsen, and Jim Torresen. 2017. Deep Models for Ensemble Touch-Screen Improvisation. In *Proceedings of AM '17, London, United Kingdom, August 23–26, 2017,* 4 pages.
https://doi.org/10.1145/3123514.3123556

1 INTRODUCTION

The proliferation of touch-screen enabled devices over the past decade has resulted in many creative digital musical instrument designs. One appealing aspect of these devices is that they are often portable and suggest group performances among friends. Several researchers have explored how these devices can be used in ensemble situations such as Wang et al.'s *MoPho* mobile phone orchestra [15], Schiemer's *Pocket Gamelan* works [9], and Snyder's marching band of DIY mobile device powered instruments [11]. These works demonstrate the very wide creative space of mobile computer music with resulting applications frequently appealing to non-expert users as well [2, 14].

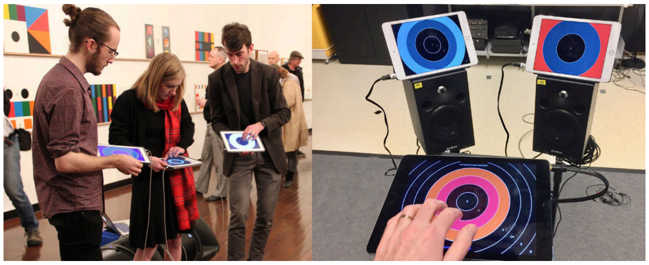

Figure 1: Data from more than 150 collaborative sessions (left) has been used to create a neural model of ensemble interaction focused on touch gestures, and a prototype system that accompanies a live performer (right).

In previous research, Martin and Gardner [5] captured a data set of more than 150 collaborative performances on mobile instruments as part of a project exploring the design and impact of networked mobile music interfaces, e.g., the left side of Figure 1. This data set included both raw touch data and time series of interpreted touch gestures sampled at a fixed interval. In the present research, we use this data set to train an artificial neural network (ANN), called *gesture-RNN*, that imitates the behaviour of the ensemble based on the input of one user. We also present an application for the ANN in a system to automatically accompany individual users of a mobile music app, thus extending the experience of collaborative creativity to users and situations where a live performance would not be possible.

Many improvised styles (e.g., jazz), have ensemble interactions based on well-defined rules. Machine learning systems such as the *Reflexive Looper*[8] and *MySong* [10] take advantage of these rules to create appropriate accompaniments. In the present research, the improvised performances were

AM '17, August 23–26, 2017, London, United Kingdom

"free", that is, performers were allowed to perform in any way they wished within the constraints of the touch-screen instruments. While it is clear that the performances contained interaction between players, it is not clear how this occurred. Thus prediction of this interaction can only be learned from the data, without assistance from music theory.

It has previously been shown that long short-term memory (LSTM) recurrent neural networks (RNNs) can be used for composing blues music [1], folk tunes [12], as well as polyphonic music [13]. These networks have not, so far, been applied to free-form improvised music using new interfaces such as touch-screens. In this work, we demonstrate that such networks can learn and generate these ensemble interactions. We also introduce a real-time performance application using our ensemble interaction RNN.

Interpreting Touch Data

Collaborative touch-screen interaction data was obtained from a publicly available corpus of performances [7]. This corpus includes data from more than 150 performances from rehearsals, concerts, installations, and demonstrations. For the present research, only ensemble improvisation data from this corpus was used. This consisted of 72 improvised ensemble performances with a total time of 9.8 hours.

Previous work [6] has defined a method for interpreting raw touch information (i.e., the time and location of taps and swipes) as a sequence of continuous touch gestures such as "slow taps", "small swirls", and "fast swipes". The resulting sequences can be analysed to examine the structure of performances.

Working with this high-level data has advantages and disadvantages. The state space is small as there are only 9 gestures in the vocabulary, and these are sampled at a regular rate (1Hz) which reduces the complexity of the data. Plots of these gesture sequences resemble graphical scores, and changes in performance style across the ensemble can be seen at a glance. The gesture sequences, however, do not have a one-to-one mapping with interaction in the app (e.g., tapping in any part of the screen could be interpreted as the "slow taps" gesture) so sonification of new sequences requires further processing into appropriate touch data.

2 ANN MODEL FOR ENSEMBLE PERFORMANCES

Our neural network model, gesture-RNN, is designed to generate an ensemble score of gesture sequences based on the real improvisation of one player. Two versions of the gesture-RNN were trained corresponding to different ensemble situations: duets and quartets. Each network uses the same architecture of a recurrent neural network (RNN) with long short-term memory (LSTM) cells. This architecture mirrors the *folkRNN* system described by Sturm et al. [12] and consists of three layers of 512 LSTM cells followed by a fully-connected softmax layer. In each case, the networks were trained with mini-batches of 64 examples of 120 time steps using the Adam optimiser with a learning rate of 0.0001. Each example corresponded to a 2-minute excerpt of a real performance. The networks were implemented in TensorFlow and were trained on an Nvidia GeForce GTX 1080 GPU. Source code for gesture-RNN is available at: https://doi.org/10.5281/zenodo.834267

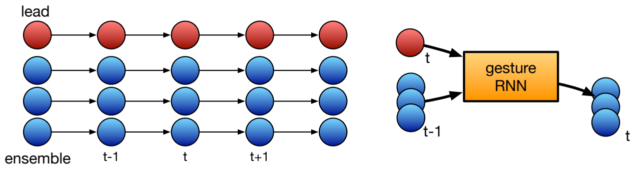

Figure 2: The gesture-RNN was trained to predict the ensemble's gestural states given the current lead state and previous ensemble states.

Representing Ensemble Behaviour

The gesture-RNNs were trained to output the gestural states of the ensemble, given input of the current lead player's state and the state of the ensemble at the previous time step. This arrangement is illustrated in Figure 2. As the training data consisted of completely free improvisations, any player could be taken as the lead, so several training examples could be generated from a single slice of a performance.

In the data set, each performer's actions are represented by one of nine touch gestures (including "nothing"); these are stored as integers in the interval $[0, 8]$. We define a simple encoding $\mathbb{N}_9^n \to \mathbb{N}_{9^n}$ to represent multiple performers as one positive integer as follows. Given a set of n performer states $g_1, g_2, \ldots, g_n$, where the g_i are integers in $[0, 8]$, the states can be encoded as: $g_1 9^0 + g_2 9^1 + g_3 9^2 + \ldots + g_n 9^{n-1}$, which is an integer in $[0, 9^n - 1]$. This encoding allows any ensemble state or combination of lead player and ensemble states to be represented as a unique integer and thus as a one-hot encoding on the inputs and outputs of the neural network.

Duet

The simplest ensemble configuration is a duet between two performers. In this model, the input for the network is the current gesture of the lead performer and the previous gesture of the second player (81 classes) while the output is the current gesture of the second player (9 classes). Training data was constructed by taking each performer as leader and constructing examples for each other player present in the performance. All ensemble performances from the corpus were treated in this manner and each sequence of 120 states was taken as a separate training example resulting in a total of 319948 examples. The network was trained for 30 epochs

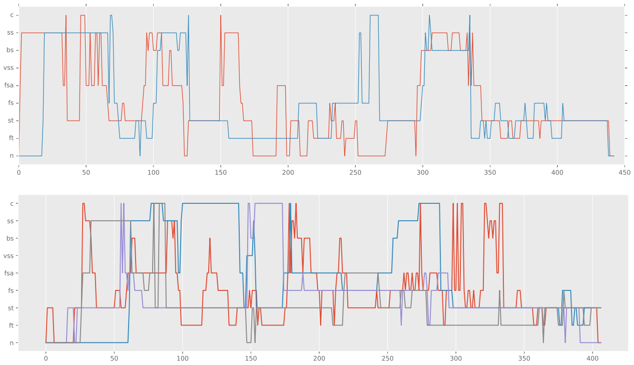

Figure 3: A generated duet (above) and quartet (below) where one lead performer (shown in red) has been accompanied by ANN-generated performers. These results show some realistic ensemble interaction.

taking 8.13 hours. This reduced the training loss (mean cross-entropy of predictions and labels over a batch) from 2.19 to 0.07.

The upper part of Figure 3 shows an example output from the network with a lead player taken from a real performance shown in red and a generated duo partner shown in blue. The generated player's sequence was primed with the "nothing" gesture in the first state. The generated player shows a similar complexity to the lead player and appears to mimic the leader's gestures at some points.

Quartet

Performances with four players were the most numerous configuration in the corpus. In this model the input for the network was the current gesture of lead player and previous gesture of the three other players (6561 classes). The output was the predicted gestures for the three others in the group (729 classes).

Training data was constructed similarly to the duet model, with each performer in the group considered as a lead player. As the ordering of the other three players was significant in the encoding (but not in the actual performance), a separate training example was constructed for each permutation of players. The quartet model gesture-RNN was trained on the whole data set of 33 quartet performances with total length of 5.28 hours corresponding to 19023 gesture state measurements. The ANN was trained for 30 epochs on a corpus of 361560 examples reducing training loss from 6.59 to 0.07; training time was 9.42 hours.

An example of a generated ensemble performance is shown in the lower part of Figure 3. In this figure, the red line shows a real gesture state sequence taken from a non-quartet improvisation (not used to train the ANN), and the other three lines show generated state sequences primed with the "nothing" gesture.

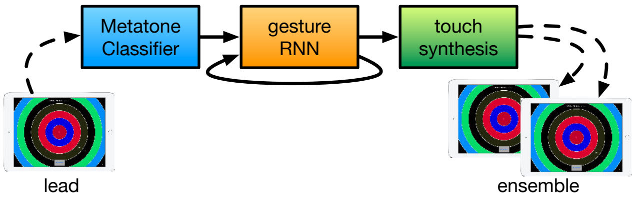

Figure 4: The live performance system classifies gestures from a live performer, generates ensemble gestures, then plays back synthesised touch events.

3 LIVE PERFORMANCE SYSTEM

The quartet gesture-RNN model was integrated into a live performance system that sonifies the ensemble. This system allows a live performer to interact with the RNN's outputs as if it was a live group. The system was created using an iPad app, PhaseRings [4], and an ensemble director agent, Metatone Classifier [3], that have been explored in previous research but extended here to incorporate the gesture-RNN's quartet model and synthesise audio outputs for the gesture sequences produced by the generated ensemble parts.

The present system is illustrated in Figure 4. One performer improvises with the PhaseRings app. The performer's gestures are classified once every second by Metatone Classifier. This gesture is used as the lead performer input for the gesture-RNN. The ensemble inputs for this RNN are initialised to the nothing gesture at the beginning of the performance. Output gestures from the RNN are sonified in the PhaseRings app running on three other iPads. An extension to Metatone Classifier generates sequences of touch events that are sent to the ensemble iPads in response to each signal from the gesture-RNN. These touch sequences are produced concatenatively from 5-second chunks of touch data recordings from live performances with PhaseRings and other iPad apps. The chunks have been labelled with their classified gesture and a random chunk is chosen and streamed to each ensemble iPad every five seconds or whenever a different gestural state is received from the RNN.

So far, this live performance system has only been used within our lab, as shown in Figure 5. The lead iPad is held by the performer. The three ensemble iPads are connected to separate monitor speakers such that the performer can hear them clearly as well as see the screens where the synthesised touches are visualised. While PhaseRings was not originally designed to playback touch interactions, the present system is based on real performance data so the output from each iPad sounds convincingly human. The app and agent system

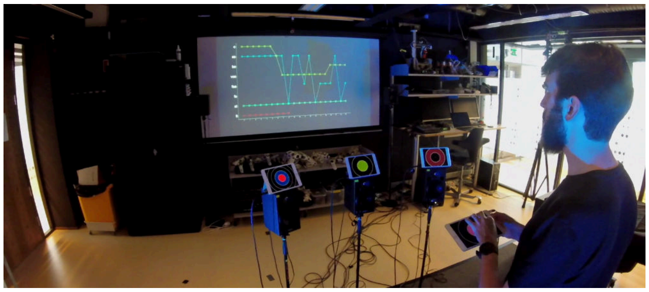

Figure 5: A performance with the live performance system, a video is available at https://doi.org/10.5281/zenodo.831910

ensures that sounds created in the ensemble are harmonically related, but since the timing of notes between iPads are not connected, performances with the system tend towards free rhythm. We envision that the present system could be used in performances or installations that explore this neural model of ensemble interaction. In this setup the separate iPads and speakers embody the simulated ensemble and allow the gesture-RNN's contribution to the performance to be examined. Other performance situations, such as representing the ensemble performers within a single app, may also be possible, and might allow individual performers to enjoy ensemble-like experiences.

4 CONCLUSIONS AND FUTURE WORK

This paper reports on work-in-progress towards a practical ANN model of ensemble interaction for touch-screen performances. Unlike other musical performance ANNs, the gesture-RNN creates transcriptions of high-level gestures rather than individual notes and generates an ensemble response to the gestures of a given lead performer. Visualising the results suggests that realistic ensemble interactions are being generated; however, further work needs to be done to evaluate the model in terms of accuracy and creativity, and to justify choice of hyper-parameters. An interactive system has been created that allows the gesture-RNN to generate responses to a live performer and then sonifies the results in real-time. This system could be used in live performances, installations, as well as to help evaluate the gesture-RNN model against more traditional sequence models.

Acknowledgements. This work is supported by The Research Council of Norway as a part of the Engineering Predictability with Embodied Cognition (EPEC) project, under grant agreement 240862.

REFERENCES

[1] Douglas Eck and Jürgen Schmidhuber. 2007. *A First Look at Music Composition using LSTM Recurrent Neural Networks.* Technical Report IDSIA-07-02. Instituto Dalle Molle di studi sull' intelligenza artificiale, Manno, Switzerland.

[2] Robert Hamilton, Jeffrey Smith, and Ge Wang. 2011. Social Composition: Musical Data Systems for Expressive Mobile Music. *Leonardo Music Journal* 21 (2011), 57–64. https://doi.org/10.1162/LMJ_a_00062

[3] Charles Martin. 2016. Metatone Classifier: Research Prototype. Git Repository. (2016). https://doi.org/10.5281/zenodo.51712

[4] Charles Martin. 2016. PhaseRings v1.2.0. Git Repository. (2016). https://doi.org/10.5281/zenodo.50860

[5] Charles Martin and Henry Gardner. 2016. A Percussion-Focussed Approach to Preserving Touch-Screen Improvisation. In *Curating the Digital: Spaces for Art and Interaction*, David England, Thecla Schiphorst, and Nick Bryan-Kinns (Eds.). Springer International Publishing, Switzerland. https://doi.org/10.1007/978-3-319-28722-5_5

[6] Charles Martin, Henry Gardner, and Ben Swift. 2015. Tracking Ensemble Performance on Touch-Screens with Gesture Classification and Transition Matrices. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME '15)*, Edgar Berdahl and Jesse Allison (Eds.). Louisiana State University, Baton Rouge, LA, USA, 359–364. http://www.nime.org/proceedings/2015/nime2015_242.pdf

[7] Charles Martin, Ben Swift, and Henry Gardner. 2016. metatone-analysis v0.1. (2016). https://doi.org/10.5281/zenodo.51710

[8] François Pachet, Pierre Roy, Julian Moreira, and Mark d'Inverno. 2013. Reflexive Loopers for Solo Musical Improvisation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2205–2208. https://doi.org/10.1145/2470654.2481303

[9] Greg Schiemer and Mark Havryliv. 2007. Pocket Gamelan: Interactive mobile music performance. In *Proceedings of Mobility Conference 2007: The 4th International Conference on Mobile Technology, Applications and Systems: (IS-CHI 2007)*. Research Publishing, 716–719.

[10] Ian Simon, Dan Morris, and Sumit Basu. 2008. MySong: Automatic Accompaniment Generation for Vocal Melodies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, New York, NY, USA, 725–734. https://doi.org/10.1145/1357054.1357169

[11] Jeff Snyder and Avneesh Sarwate. 2014. Mobile Device Percussion Parade. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME '14)*. Goldsmiths, University of London, London, UK, 147–150. http://www.nime.org/proceedings/2014/nime2014_542.pdf

[12] Bob L. Sturm, Jo ao Felipe Santos, Oded Ben-Tal, and Iryna Korshunova. 2016. Music Transcription Modelling and Composition Using Deep Learning. In *Proceedings of the 1st Conference on Computer Simulation of Musical Creativity*. University of Huddersfield, UK.

[13] Christian Walder. 2016. Modelling Symbolic Music: Beyond the Piano Roll. In *Proceedings of The 8th Asian Conference on Machine Learning*, Robert J. Durrant and Kee-Eung Kim (Eds.), Vol. 63. PMLR, 174–189. http://proceedings.mlr.press/v63/walder88.html

[14] Ge Wang. 2014. Ocarina: Designing the iPhone's Magic Flute. *Computer Music Journal* 38, 2 (2014), 8–21. https://doi.org/10.1162/COMJ_a_00236

[15] Ge Wang, Georg Essl, and Henri Penttinen. 2014. The Mobile Phone Orchestra. In *The Oxford Handbook of Mobile Music Studies*, Sumanth Gopinath and Jason Stanyek (Eds.). Vol. 2. Oxford University Press, Oxford, UK, 453–469. https://doi.org/10.1093/oxfordhb/9780199913657.013.018