PREDICTING EXPRESSIVE DYNAMICS IN PIANO PERFORMANCES USING NEURAL NETWORKS

Sam van Herwaarden
Austrian Research Institute for AI
samvherwaarden@gmail.com

Maarten Grachten
Austrian Research Institute for AI

W. Bas de Haas
Utrecht University
w.b.dehaas@uu.nl

ABSTRACT

This paper presents a model for predicting expressive accentuation in piano performances with neural networks. Using Restricted Boltzmann Machines (RBMs), features are learned from performance data, after which these features are used to predict performed loudness. During feature learning, data describing more than 6000 musical pieces is used; when training for prediction, two datasets are used, both recorded on a Bösendorfer piano (accurately measuring note on- and offset times and velocity values), but describing different compositions performed by different pianists. The resulting model is tested by predicting note velocity for unseen performances. Our approach differs from earlier work in a number of ways: (1) an additional input representation based on a local history of velocity values is used, (2) the RBMs are trained to result in a network with sparse activations, (3) network connectivity is increased by adding skip-connections, and (4) more data is used for training. These modifications result in a network performing better than the state-of-the-art on the same data and more descriptive features, which can be used for rendering performances, or for gaining insight into which aspects of a musical piece influence its performance.

1. INTRODUCTION

Music is not performed exactly the way it is described in score: a performance in which notes occur on a regular temporal grid and all notes are played equally loud is often considered dull. Depending on the instrument, performers have different parameters they use for modulating expression in their music [14]: time (timing, tempo), pitch, loudness and timbre. For some of these parameters composers add annotations to musical score describing how they should be varied, but for a large part performers are expected render the score according to tacit knowledge, and personal judgment. This allows performers to imbue on a performance their personal style, but this is not to say that music performance is arbitrary—it is often clear which interpretations are (not) musically appropriate.

This article describes a number of modifications to the method for modeling expressive dynamics proposed by Grachten & Krebs [7], and is based on the MSc thesis work described in [17]. We show that, with an additional input representation and a different set-up of the machine learning approach, we achieve a statistically significant improvement on the prediction accuracy achieved in [7], with more descriptive features. Our achieved performance is also comparable with the work in [8]. In the following sections we first summarize previous work in this area, followed by an overview of the used machine learning architecture. We then describe the experiments, the results and the relevance of the findings.

2. PREVIOUS WORK

Two important aspects of music that affect the way it is to be performed are the musical structure, and the emotion that the performance should convey [13]. The last decades different methods for analyzing the structural properties of a piece of music have been proposed (e.g. [12, 15]), where the analysis tends to stress the relationship between structure on a local level (elements of pitch and rhythm) and their effect on the melodic expectancy of a listener. Emotional charge conveyed by a piece is more abstract and variable: trained musicians can play the same piece conveying different emotions, and in fact these emotions can be identified by listeners [5].

Because musical structure can be studied through inspection of the musical score, computational models of musical expression tend to focus on this. A number of different computational models of expression have been developed earlier, studying different expressive parameters (e.g. [1, 4]). Many models are rule-based, where the rules describing how expression should be applied are often hand-designed. Other models still focus on rules, but automatically extract them from performance data (e.g. [11, 18]). A performance model can also be based on the score annotations for the relevant parameter provided by the composer, as in [8] which uses information on note pitch, loudness annotations and other hand-crafted features.

Some recent studies model regularities in musical sequences using unsupervised techniques [2, 16], in the context of musical sequence prediction. Grachten & Krebs [7] apply unsupervised learning techniques to learn features from a simple input representation based on a piano roll

representation of the symbolic score, in the context of predicting musical expression. The resulting learned features then describe common patterns occurring in the input data, which can be related to concepts from music theory and used for prediction of expressive dynamics. By using a simple input representation and network, the model remains relatively transparent with regard to its inner workings. It is shown that Restricted Boltzmann Machines (RBMs) learn the most effective model, and in this paper, we build on that approach.

An RBM is a type of artificial neural network, particularly suitable for unsupervised learning from binary input data. During training it learns a set of features that can efficiently encode the input data. The features are used to transform the input data non-linearly, which can be useful for further (supervised) learning. For a detailed explanation of RBMs the reader is referred to for example [9, 10].

3. ARCHITECTURE

Figure 1 illustrates the setup of the network we use. As input the network sees the music data in two different representations: the score-based note-centered representation first developed by [7] and the new loudness-based velocity-history representation. The input data is transformed through a series of hidden unit activations (RBM feature activations) in $\mathbf{L_1}$, $\mathbf{L_2}$ and $\mathbf{L_3}$. These feature activations are then used to estimate the output (normalized velocity). As is typical with neural networks, the model is blind to the meaningful ordering of the input nodes (we could change the ordering without affecting the results).

The set-up is different from that in [7] in a number of ways: (1) an additional input representation based on a local history of velocity values is used, (2) the RBMs are trained for sparse activations, (3) network connectivity is increased with skip-connections (i.e. $\mathbf{w_1}$ and $\mathbf{w_2}$ in Figure 1 can be used simultaneously), and (4) more data is used for training. The following sections cover these changes in more detail. First, we describe the data available for developing the model. We then describe the way these data are presented to our model as input and output, and finally the process of training and evaluating our model.

3.1 Available data

Data from a number of sources is used for the experiments in this paper. We have *score* data, which describes musical score in a piano-roll fashion, and we have *performance* data, based on recordings from a computer-controlled Bösendorfer piano. For the performance data, accurate note on- and offsets are available as well as velocity values, and these values have been linked to corresponding score data. For all available performance data, score data is also available, the converse does not hold.

A number of (MIDI) score datasets is used: the JSB Chorales, [1] some MuseScore pieces, [2] the Mutopia database, [3] the Nottingham database, [4] the Piano-midi archive [5] and the Voluntocracy dataset [6]. These datasets are used during unsupervised learning with the note-centered representation only. The performance datasets we use have been developed at the Austrian Research Institute for AI (OFAI). One dataset contains performance data of all Chopin's piano music played by Nikita Magaloff [3] (~ 300.000 notes in 155 pieces), the other contains all Mozart piano sonatas, performed by Roland Batik [18] (~ 100.000 notes in 128 pieces). These datasets have been used both for unsupervised and supervised learning.

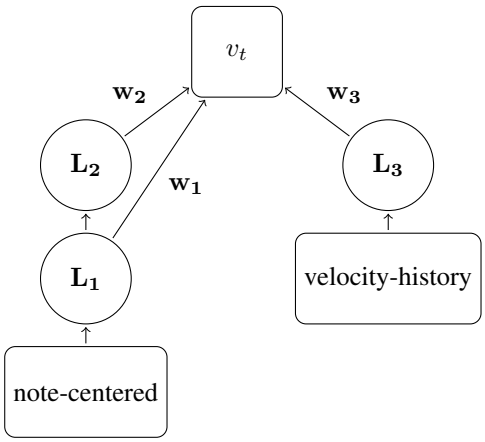


Figure 1: The used architecture. The rounded squares correspond to in- and outputs, the circles to layers of hidden units trained as Restricted Boltzmann Machines. $\mathbf{w_1}$ through $\mathbf{w_3}$ are the weights used to predict v_t based on the hidden unit activations in hidden layers $\mathbf{L_1}$ through $\mathbf{L_3}$. $\mathbf{w_1}$ through $\mathbf{w_3}$ are determined with a least-squares fit.

3.2 Note-centered representation

Score data is input into the network in one form in the *note-centered representation*, which is based on a piano-roll representation. For every note in a musical score, an input sample is generated with this note in the center, as illustrated in Figure 2b. The horizontal axis corresponds to score time and covers a span of 3 beats before the onset of the central note to 3 beats after the onset. Each beat is further divided into 8 equal units of time (effectively each column in the input corresponds to a 32nd note), and longer notes are wider. The vertical axis corresponds to relative pitch compared to the central note, and covers a span of -55 to $+55$ semi-tones. To allow the representation to distinguish between separate notes of the same pitch played consecutively, and a single long note at that pitch, note durations are represented as their score duration minus 32nd note duration (this was also done in [7]).

This approach is the same as the *duration coding* approach used in [7] with two exceptions: they experimented with time-spans of 1, 2 and 4 beats (with very small dif-

[1] www.jsbchorales.net
[2] www.musescore.org
[3] www.mutopiaproject.org
[4] www.chezfred.org.uk/University/music/database.htm
[5] www.piano-midi.de
[6] www.voluntocracy.org

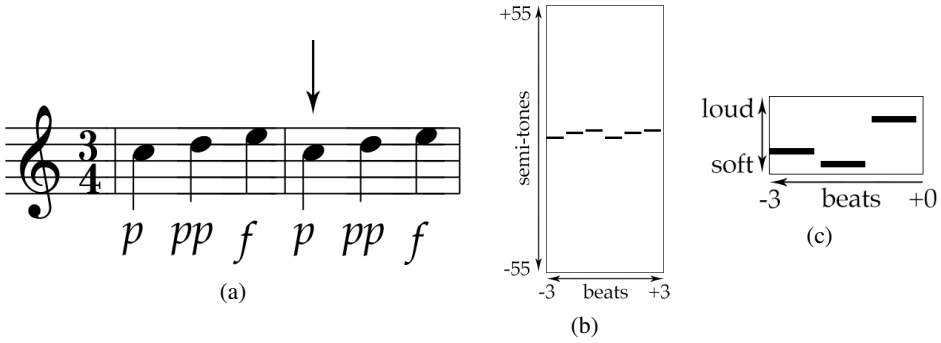


Figure 2: A short piece of score, and resulting network input for the note indicated by the arrow: (a) shows the score, where the annotations should be interpreted as *performed* loudness, not as annotated loudness directives, (b) shows the note-centered representation and (c) the velocity-history representation.

ferences in results between the 2 and 4 beats experiments), and used a pitch range of -87 to $+87$ semi-tones (so that always the entire piano keyboard is covered). In practice, the large pitch range is likely unnecessary and only increases the length of the network input vector (note combinations with such intervals are very rare and do not noticeably affect the learned features).

This choice of representation makes our system insensitive to absolute pitch: if all input notes are transposed by a few semi-tones in the same direction, the generated input samples will be identical. This also allows the system to learn about harmony based on relative pitch: for example certain chords will typically be represented in the same way regardless of their root tone. No additional information on absolute note pitch was included, to keep the model simple.

3.3 Velocity-history representation

When analyzing expressive parameters in existing performances, it is interesting to not only take into account direct harmonic and rhythmic structure around a note as is done with the note-centered representation, but also effects in continuity of musical phrases: for example, in many cases note loudness increases or decreases gradually over a number of notes. The precise accentuation of a note is than affected by the accentuation of preceding notes.

Our *velocity-history representation* is designed to encode this kind of information. Figure 2c illustrates this representation. Conceptually, it is similar to the note-centered representation, with a few differences: the vertical axis now represents relative velocity (normalized with respect to the mean μ and standard deviation σ of the velocity in a piece, where the range from $\mu - 2\sigma$ to $\mu + 2\sigma$ is quantized into 12 discrete values), and the horizontal axis corresponds to the time preceding the current note (ranging from note onset -3 beats to note onset $+0$).

The velocity-history representation uses information from an actual performance during prediction. In a sense, the system is asked to predict the continuation of a musical phrase: given that the last notes were played in a certain way, how will the next note be played? When using this representation, experiments with our model aim to *explain* how a note is performed in an existing performance, rather than *predict* it for a new piece of bare score (an actual performance needs to be available).

3.4 Velocity normalization

Since we use semi-supervised learning, at some point we need target values accompanying our input representations. We have exactly one sample for each note, and we are studying dynamics, so the logical parameter to base these target values on is note velocity. However, the different pieces described in our data have fairly diverse characteristics when it comes to dynamics. Some pieces are performed louder on average, or have stronger variations in dynamics. In this study we have chosen to focus on local effects within a single piece, and not so much on differences between pieces. For this reason we normalize our velocity target values so they have zero-mean and unit standard-deviation within a piece (we use these values both for supervised learning and for generating the velocity-history representation). This is slightly different from the normalization used in [7], where normalization was only used to obtain zero-mean within a piece.

3.5 Training and evaluation

The process of developing and testing the network can be separated into three phases: unsupervised learning, supervised learning and performance evaluation. We will now describe these in more detail.

3.5.1 Unsupervised learning

During unsupervised learning, we train only hidden layers $\mathbf{L_1}$ through $\mathbf{L_3}$. The layers are trained as RBMs on the full set of score data in the note-centered and velocity-history representations, where $\mathbf{L_1}$ and $\mathbf{L_3}$ are trained on the input representations directly, and $\mathbf{L_2}$ is trained on the feature activations in $\mathbf{L_1}$.

In the note-centered representation samples consist of 5280 binary input values. $\mathbf{L_1}$ is trained with 512 hidden units (ensuring a significant bottleneck in the network), and $\mathbf{L_2}$ contains fewer hidden units again: 200 units. In the velocity-history representation samples consist of 288 input values, these are encoded in 120 hidden units in $\mathbf{L_3}$.

We enforce sparse coding in the network, using the method proposed in [6], which allows us to not only control the average activation of hidden units in the network, but also the actual distribution of activations: we can force the RBM to represent each sample as a number of highly active features, improving inspectability.

3.5.2 Supervised learning

For supervised learning we use a simple approach: given the transformation of an input sample by $\mathbf{L_1}$ to $\mathbf{L_3}$, we fit the hidden unit activations in these layers to the corresponding v_t (normalized velocity) values using least-squares. Exploratory experiments suggested that more advanced techniques do not yield much better results. Thus, $\mathbf{w_1}$ through $\mathbf{w_3}$ simply define a linear transformation from the features activations to a prediction of the normalized velocity.

3.5.3 Performance evaluation

To evaluate the performance of our model we use a leave-one-out approach: we cycle through all the pieces in the performance data, where every time a particular piece is left out during supervised learning, after which the trained network is used to predict the expressive dynamics of the left-out piece. The quality of the prediction is then quantified using the R^2 measure (coefficient of determination). As mentioned before, the full set of data is used during unsupervised learning – because the objective function optimized during this phase has no relation to the velocity targets, we believe that this is an acceptable approach. As the final score after cycling through the whole dataset in this fashion, we use the weighted average R^2, where the number of notes in a piece is used as its weight.

4. EXPERIMENTS

In our experiments we vary two parameters: network connectivity, and training/testing datasets. Other experiments were also done but are not described in this paper, for these the interested reader is referred to [17].

4.1 Network connectivity

Different parts of our model describe information concerning different aspects of the input data. The note-centered representation corresponds to rhythmic and harmonic structure of the score surrounding a note, while the velocity-history representation relates more closely to expressive phrases. This distinction continues through the layers of feature activations. To get an impression of how strongly the expressive variation in velocity data corresponds to these different aspects, we experimented with the different layers in isolation and together. We will refer to the network configurations by the layers that were used during training and prediction, i.e. $\mathbf{L_{1,2}}$ means both of the layers on top of the note-centered representation were used, and $\mathbf{L_3}$ was not. Another way to see this would be that $\mathbf{w_3}$ is constrained to be a matrix of only 0's.

	no vel. inf.			**with vel. inf.**	
	$\mathbf{L_1}$	$\mathbf{L_{1,2}}$	$\mathbf{L_2}$	$\mathbf{L_3}$	$\mathbf{L_{1,2,3}}$
M. $\rightarrow$ **M.**	.202	<u>.207</u>	.191	.315	.470
B. $\rightarrow$ **B.**	.366	.376	.357	.236	.532
B. $\rightarrow$ **M.**	.132	.126	.125	.286	.386
M. $\rightarrow$ **B.**	.291	.295	.283	.209	.457
All $\rightarrow$ **M.**	.198	.203	.186	.313	.466
All $\rightarrow$ **B.**	.341	.350	.329	.222	.503

Table 1: $\bar{R}^2$ scores obtained on the test data. $X \rightarrow Y$ indicates the model was trained on X and tested on Y, where **M.** is the Magaloff and **B.** the Batik dataset. Experiments with velocity information (vel. inf.) use the velocity-history representation as input. We use the underlined result for comparison with previous work ([7] and [8]).

4.2 Training datasets

Experimenting with different sets of training data is interesting for several reasons. One is that from a musicological perspective, the structure of music of different styles can be quite different. As an extreme example, a system trained on Jazz music would not be expected to reliably predict performances of piano music by Bach. Another reason is that we can use combinations of datasets to test the validity of our model: if a model trained on music from one set of recordings, still performs well on another set of recordings, this can give us some confidence that our model has learned something about music in a general sense, and not just about the particular dataset.

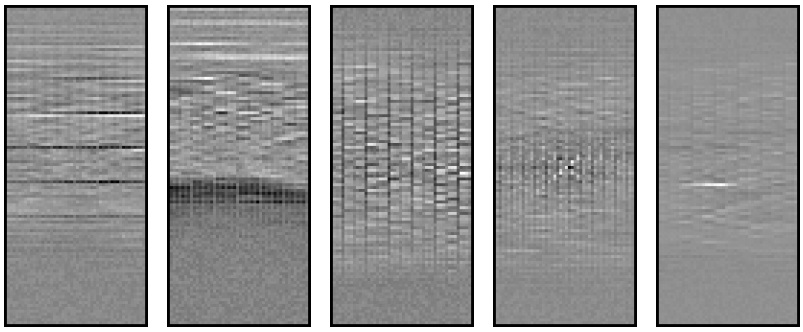
As mentioned before, we use two datasets: one describing performances of Chopin music and the other Mozart music. In all cases, during testing we kept the datasets separate. However, we varied the set of data used for training: we trained on the same dataset as used for testing, we trained on one dataset and tested on the other, and we tested a model trained on all data.

5. RESULTS

Table 1 lists the results obtained with our model. The model is more successful explaining the variance in the Batik (Mozart) data than in the Magaloff (Chopin) data – one possible explanation for this is that Chopin's music (from the Romantic period) has much more extreme variations in expression than Mozart's music (from the Classical period). It seems reasonable that a performance with more dynamic variation is harder to predict.

When comparing the different architectures, most information used by our model is encoded in $\mathbf{L_1}$ and $\mathbf{L_3}$. $\mathbf{L_2}$ has less predictive value than $\mathbf{L_1}$, and the score only improves by a little bit when these two layers are used together (suggesting there is a large amount of overlap in the information they encode). $\mathbf{L_3}$, which is based on the velocity-history representation (which was not used in [7]) clearly contains a lot of information.

Interestingly, $\mathbf{L_3}$ contains most relevant information for



(a) (b) (c) (d) (e)

Figure 3: Some hand-selected features from $\mathbf{L_1}$ that are representative for the types of patterns learned from the note-centered representation (see Figure 2b). Dark values correspond to negative weights, light values to positive weights.

the Magaloff data, and $\mathbf{L_1}$ for the Batik data. This could be due to the difference between music from the Romantic period and that from the Classical period: $\mathbf{L_1}$ contains more information about harmony, whereas $\mathbf{L_3}$ contains more information about the expressive 'flow' of the piece.

Training on a single dataset has a positive effect on the prediction scores. This is likely due to the fact that the datasets are of a different nature in terms of musical style, and if we would want to predict performance parameters for a Mozart piece, training on Chopin music will not provide our model with the relevant 'know-how'. This is also illustrated by the cross-training experiments, where we trained on one dataset and tested on the other: a drop in performance of around 0.08 in all cases is observed. Still, also a relatively large amount of the predictive capability remains, providing some confidence that our model generalizes over different datasets to some extent.

Because the velocity-history representation requires detailed performance data for predictions, we use the results from our $\mathbf{L_{1,2}}$ experiments when comparing our results to earlier work which does not use performance data. In [7] the best obtained $\bar{R}^2$ score on the Magaloff data is .139, using a single dense RBM layer with 1000 hidden units (similar to our $\mathbf{L_1}$ model). Our $\mathbf{L_{1,2}}$ model achieved an $\bar{R}^2$ of .207 on the same dataset. To keep statistical testing simple, we tested the statistical significance of the difference in *unweighted* average R^2 of our model and the model in [7] using a Wilcoxon signed rank test. We chose the Wilcoxon test because the underlying distribution of the R^2 data is unknown. We found that the unweighted average R^2 of .199 of our $\mathbf{L_{1,2}}$ model is significantly different from the unweighted average R^2 of .121 of the model in [7] ($W = 11111, p < 2.2 \cdot 10^{-16}$). In [8], the maximal obtained prediction accuracy on the Magaloff dataset is an $\bar{R}^2$ of .188. This model uses information our models have no access to, most importantly dynamic score annotations. Nevertheless, with an $\bar{R}^2$ of .207 our $\mathbf{L_1 L_2}$ model again seems more successful even though it does not take such annotations into account. [7] When we do use performance data, the difference becomes more pronounced: our $\mathbf{L_{1,2,3}}$ model obtains an $\bar{R}^2$ of .470 on the Magaloff data.

Something interesting to mention here is that in [17] we also experimented with limiting training data to a particular genre (i.e. training only on Nocturnes). These experiments suggested that the velocity-history representation encodes some genre-specific information, however due to space constraints we do not cover these results further here.

6. DISCUSSION

We discuss two properties of our model: the features that were learned from the musical data, and the performance achieved during prediction. Figure 3 illustrates a number of hand-selected features that have been learned from the note-centered representation, which were chosen to give an impression of the variety of learned features. Compared to the features learned by [7], there is a larger variety of features, where features represent sharper patterns.

6.1 Learned features

Figure 3 illustrates some of the learned features. The displayed features were selected so as to give the reader an impression of the diversity of the learned features . From a musicological perspective, it is interesting to see that there seem to be some remarkable patterns relating the features to music theory. The features learned from the velocity-history representation are harder to interpret musicologically, these are not further discussed in this paper.

Figure 3a shows clear horizontal banding, where interestingly the bands are exactly 12 rows apart – this corresponds to octaves. The feature in some locations displays a strong contrast between pitches one semi-tone apart, which is related to dissonance.

A common pattern is illustrated in Figure 3b, with a dark (inhibitive) band above or below a lighter region. This type of feature is also described by Grachten & Krebs [7], who argue this can be regarded as an accompaniment versus melody detector: the illustrated feature is strongly inhibited by notes in a sample that are below the central note, meaning that the feature activates more readily for bass notes. The opposite type of feature, with inhibitive regions above and excitatory regions below the central note (not

[7] To perform the statistical test, detailed results from [7] were kindly provided by the authors. For the work in [8] these results were unfortunately unavailable, meaning we could not perform the same statistical analysis with this result.

shown here), is active with a high probability for melody notes, where surrounding notes have lower pitch.

Another common pattern is the vertical banding illustrated by Figure 3c. There is some variation in the offset of the vertical bands from the edges (their phase) and how close they are together (their period). These features can convey information on the pace in the current part of the piece (predominantly short or long notes) and the temporal position of the note with respect to the beat.

A few features also display diagonal banding as illustrated by Figure 3d, although these are relatively rare. Still, we hypothesize that with these our model can deduce whether the central note is in an ascending or descending sequence.

A final common pattern is that in Figure 3e, with a sharp white band corresponding to a note at a certain relative pitch and time from the central note. It seems reasonable to suggest that these can be related to particular melodic steps – changes from one note to another with a particular relative pitch and timing.

6.2 Model performance

The performance of our model is an improvement compared to earlier work, particularly when the goal is to *explain* the structure of an existing performance rather than predict a performance for a new piece of score – in the former situation the velocity-history representation can be used to good effect. Still, when considering a purely predictive context (using no velocity information), an R^2 of around 0.2 leaves room for improvement. There is of course a practical limit in terms of what score can be obtained: even the same pianist might not play a piece in exactly the same way on different occasions, meaning that an R^2 close to 1.0 cannot be expected. A factor that limits our model is that it considers score structure at a local level only – structure at larger timescales is not considered, nor are loudness annotations, which of course also convey a lot of information about how loudly a particular piece of score is to be played. These omissions are opportunities for further work: including these components could improve performance further, for example loudness annotations could be included similarly to what was done in [8].

7. CONCLUSIONS

We showed that neural networks trained on relatively raw representations of musical score and musical performances can be used to predict expressive dynamics in piano performances. This was done before in [7], but we changed the learning architecture (using sparse RBMs and skip-connections), and developed a new input representation, resulting in better predictions and clearer features. We also showed that our model generalizes well to datasets on which it was not trained.

8. ACKNOWLEDGEMENTS

The research described in this article was sponsored by the Austrian Science Fund (FWF) under project Z159 (Wittgenstein Award), and by the European Commission under the projects Lrn2Cre8 (grant agreement no. 610859), and PHENICX (grant agreement no. 601166). The research was part of an MSc project resulting in a thesis [17], the contents of which overlap to some extent with those in this paper. W. Bas de Haas is supported by the Netherlands Organization for Scientific Research, through the NWO-VIDI-grant 276-35-001.

9. REFERENCES

[1] E. Bisesi and R. Parncutt. An accent-based approach to automatic rendering of piano performance: Preliminary auditory evaluation. *Archives of Acoustics*, 36(2):283–296, 2010.

[2] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In *Proceedings of the Twenty-nine International Conference on Machine Learning*. ACM, 2012.

[3] S. Flossmann, W. Goebl, M. Grachten, B. Niedermayer, and G. Widmer. The magaloff project: An interim report. *Journal of New Music Research*, 39(4):363–377, 2010.

[4] A. Friberg, L. Fryden, L. Bodin, and J. Sundberg. Performance rules for computer-controlled contemporary keyboard music. *Computer Music Journal*, 15(2):49–55, 1991.

[5] A. Gabrielsson and P.N. Juslin. Emotional expression in music performance: Between the performer's intention and the listener's experience. *Psychology of music*, 24(1):68–91, 1996.

[6] H. Goh, N. Thome, and M. Cord. Biasing restricted boltzmann machines to manipulate latent selectivity and sparsity. In *NIPS workshop on deep learning and unsupervised feature learning*, 2010.

[7] M. Grachten and F. Krebs. An assessment of learned score features for modeling expressive dynamics in music. *Transactions on multimedia: Special issue on music data mining*, 16(5):1–8, 2014.

[8] M. Grachten and G. Widmer. Explaining musical expression as a mixture of basis functions. In *Proceedings of the 8th Sound and Music Computing Conference (SMC 2011)*, 2011.

[9] G.E. Hinton. A practical guide to training restricted boltzmann machines. *Momentum*, 9(1):926, 2010.

[10] G.E. Hinton and T.J. Sejnowski. Learning and relearning in boltzmann machines. *MIT Press, Cambridge, Mass*, 1:282–317, 1986.

[11] H. Katayose and S. Inokuchi. Learning performance rules in a music interpretation system. *Computers and the Humanities*, 27(1):31–40, 1993.

[12] F. Lerdahl and R.S. Jackendoff. *A generative theory of tonal music*. The MIT Press, 1983.

[13] C. Palmer. Music performance. *Annual review of psychology*, 48(1):115–138, 1997.

[14] R. Parncutt. Accents and expression in piano performance. *Perspektiven und Methoden einer Systemischen Musikwissenschaft*, pages 163–185, 2003.

[15] H. Schenker. Five graphic music analyses, 1969.

[16] A. Spiliopoulou and A. Storkey. Comparing probabilistic models for melodic sequences. In *Proceedings of the 2011 European conference on Machine learning and knowledge discovery in databases - Volume Part III*, ECML PKDD'11, pages 289–304, Berlin, Heidelberg, 2011. Springer-Verlag.

[17] S. van Herwaarden. Teaching neural networks to play the piano. Master's thesis, Utrecht University, 2014.

[18] G. Widmer. Large-scale induction of expressive performance rules: First quantitative results. In *Proceedings of the International Computer Music Conference (ICMC'2000)*, pages 344–347, 2000.