# MULTIPLE VIEWPOINT MELODIC PREDICTION WITH FIXED-CONTEXT NEURAL NETWORKS

**Srikanth Cherla[1,2], Tillman Weyde[1,2] and Artur d'Avila Garcez[2]**
[1]Music Informatics Research Group, Department of Computer Science, City University London
[2] Machine Learning Group, Department of Computer Science, City University London
{srikanth.cherla.1, t.e.weyde, a.garcez}@city.ac.uk

## ABSTRACT

The *multiple viewpoints* representation is an event-based representation of symbolic music data which offers a means for the analysis and generation of notated music. Previous work using this representation has predominantly relied on $n$-gram and variable order Markov models for music sequence modelling. Recently the efficacy of a class of distributed models, namely restricted Boltzmann machines, was demonstrated for this purpose. In this paper, we demonstrate the use of two neural network models which use fixed-length sequences of various viewpoint types as input to predict the pitch of the next note in the sequence. The predictive performance of each of these models is comparable to that of models previously evaluated on the same task. We then combine the predictions of individual models using an entropy-weighted combination scheme to improve the overall prediction performance, and compare this with the predictions of a single equivalent model which takes as input all the viewpoint types of each of the individual models in the combination.

## 1. INTRODUCTION

We are interested in the computational modelling of melodies available in symbolic music data formats such as MIDI and KERN. For this purpose, we chose to work with a representation of symbolic music first proposed in [9] in relation to *multiple viewpoints for music prediction* (which we refer to here as the "multiple viewpoints representation"). The multiple viewpoints representation is an event-based representation extracted from symbolic music data where a given piece of music is decomposed into parallel streams of features, known as *viewpoint types*. Each viewpoint type is either a directly observable musical dimension such as *pitch* and *note duration*, or an abstract one derived from them such as *inter-onset interval* or *pitch contour*. In order to analyse musical structure using this representation, one can train a machine learning model on sequences of viewpoint types and apply it to tasks such as music generation [6] and classification [3, 7]. This representation has also been the focus of more recent work related to music cognition [14,17]. The novelty of this approach is in its extension of previous work in language modelling to music with an information theoretic backing which facilitates an objective evaluation of models for music prediction. Approaches based on information theory have been of interest in musicology to understand structure and meaning in music in terms of its predictability [10, 11, 13].

In the original work on multiple viewpoints [9] and that which followed [15, 21], Markov models were exclusively employed for music modelling using this framework. While this is a reasonable choice, Markov models are often faced with a problem related to data sparsity known as the *curse of dimensionality* [2]. This refers to the exponential rise in the number of model parameters to be estimated with the length of the modelled sequences. Models which employ distributed architectures such as neural networks tend to avoid this problem, as they do not require enumerating all state transition probabilities, but rather the weights of the network encode only those dependencies necessary to minimize prediction error. It was demonstrated more recently in [4] how a distributed model — the restricted Boltzmann machine, is a suitable alternative in this context. It was also suggested in [8] that neural networks might be suitable alternatives to $n$-gram models for music modelling with multiple viewpoints but no actual research in this direction has ensued.

In this paper, we first present two neural networks for modelling sequences of musical pitch. The first is a simple feed-forward neural network [20], and the second is the musical extension of the Neural Probabilistic Language Model [2] — a deeper feed-forward network with an added weight-sharing layer between the input and hidden layers. The latter was originally proposed for learning distributed representations of words in language modelling. Both models predict a probability distribution over the possible values of the next pitch given a fixed-length context as input. Their predictive performance is comparable to or better than previously evaluated melody prediction models in [4, 16]. The second network is further extended to make use of additional viewpoint types extracted from the context, as inputs for the same task of predicting musical pitch. We then combine the predictions of individual models with different viewpoint types as their respective

inputs using an entropy-weighted combination scheme to improve the overall prediction performance, and compare this with the predictions of a single model which takes as input all the viewpoint types of each of the individual models in the combination.

We begin with an overview of the multiple viewpoints representation in Section 2. This is followed by a description of the two neural networks which are used with this representation, in Section 3. Section 4 presents an evaluation of the predictive performance of the two models along with a comparison to previous work. Finally, directions for future research are outlined in Section 5.

## 2. MULTIPLE VIEWPOINT SYSTEMS

In order to explain music prediction with multiple viewpoints, the analogy to natural language is used here. In statistical language modelling, the goal is to build a model that can estimate the joint probability distribution of subsequences of words occurring in a language $L$. A statistical language model (SLM) can be represented by the conditional probability of the next word $w_T$ given all the previous ones $[w_1, \ldots, w_{(T-1)}]$ (written here as $w_1^{(T-1)}$), as

$$P(w_1^T) = \prod_{t=1}^{T} P(w_t | w_1^{(t-1)}) \; . \tag{1}$$

The most commonly used SLMs are *n-gram* models, which rely on the simplifying assumption that the probability of a word in a sequence depends only on the immediately preceding $(n-1)$ words [12]. This is known as the Markov assumption, and reduces (1) to

$$P(w_1^T) = \prod_{t=1}^{T} P(w_t | w_{(t-n+1)}^{(t-1)}) \; . \tag{2}$$

Following this approach, musical styles can be interpreted as vast and complex languages [9]. In predicting music, one is interested in learning the joint distribution of *musical event* sequences $s_1^T$ in a *musical language* $S$. Much in the same way as an SLM, a system for music prediction models the conditional distribution $p(s_t | s_1^{(t-1)})$, or under the Markov assumption $p(s_t | s_{(t-n+1)}^{(t-1)})$. For each prediction, context information is obtained from the events $s_{(t-n+1)}^{(t-1)}$ immediately preceding $s_t$. Musical events have a rich internal structure and can be expressed in terms of directly observable or derived musical features such as pitch, note duration, inter-onset interval, or a combination of two or more such features. The framework of multiple viewpoint systems for music prediction [9] was proposed in order to efficiently handle this rich internal structure of music by exploiting information contained in these different musical feature sequences, while at the same time limiting the dimensionality of the models using these features. In the interest of brevity, we limit ourselves to an informal discussion of multiple viewpoint systems for monophonic music prediction and refer the reader to [9] for a more detailed explanation.

A musical event $s$ refers to the occurrence of a note in a melody. A *viewpoint type* (or simply *type*) $\tau$ refers to any of a set of musical features that describe an event. The domain of a *type*, denoted by $[\tau]$ is the set of possible values of that type. A *basic type* is a directly observable or given feature such as *pitch, note duration, key-signature* or *time-signature*. A *derived type* can be derived from any of the basic types or other derived types. Two or more types can be "linked" by taking the Cartesian product of their respective domains, thus creating a *linked viewpoint type*. A *multiple viewpoints system* (MVS) is a set of models, each of which is trained on subsequences of one *type*, whose individual predictions are combined in some way to influence the prediction of the next event in a given event sequence. Given a context $s_{(t-n+1)}^{(t-1)}$ and an event $s_t$, each viewpoint $\tau$ in an MVS must compute the probability $p_\tau(s_t | s_{(t-n+1)}^{(t-1)})$.

In order to input the viewpoint type sequences to the neural network models, we first convert each input type value into a binary one-hot encoding. When a context event is missing or undefined, each element of the vector is initialized to $1/|S|$. When there is more than one input type, one-hot vectors corresponding to all the input types for a musical event are concatenated to obtain an input vector for that event. As we are dealing with models of fixed context-length $l$, the final input feature vector input to the model is a concatenation of $l$ such vectors. In doing so, we are effectively bypassing the need to compute a Cartesian product to link viewpoint types before using them as input to a single model which has been the practice when using $n$-gram and variable order Markov models.

Each model in an MVS relies on a different source of information (its respective input types) to make a prediction about the target viewpoint type. The accuracy of the prediction depends on how informative these input types are of the target type. It is possible to combine the information provided by different input types for possibly better predictive performance. Here, we consider two ways of doing this - *implicitly* in a single model which is trained using a set of input types, and *explicitly* by combining the probability distributions of multiple models, each of which is trained separately on a mutually exclusive subset of these input types. While the former is only a special case of what has been described so far, we provide an explanation of the latter below in Section 2.1.

### 2.1 Combining Multiple Models

It was demonstrated in [9, 15] that an entropy-weighted combination of the predictions of two or more $n$-gram or variable order Markov models typically results in ensembles with better predictive performance than any of the individual models. As it is the predicted distributions which are combined, this approach is independent of the types of models involved. Here, we briefly describe two approaches for creating such ensembles. Let $M$ be a set of models and $p_m(s)$ be the probability assigned to symbol $s \in [\tau_{tgt}]$ by model $m$, where $[\tau_{tgt}]$ is the domain of the target type. The first approach involves taking a weighted arithmetic mean of their respective predictions. This is the *mixture-*

*of-experts* combination, and is defined as

$$p(s) = \frac{\sum_{m \in M} w_m p_m(s)}{\sum_{m \in M} w_m}$$

where each of the weights $w_m$ depends on the entropy of the distribution generated by the corresponding model $m$ in the combination such that greater entropy (and hence uncertainty) is associated with a lower weight [5]. The weights are given by the expression $w_m = H_{rel}(p_m)^{-b}$, where the relative entropy $H_{rel}(p_m)$ is

$$H_{rel}(p_m) = \begin{cases} H(p_m)/H_{max}(p_m), & \text{if } H_{max}([\tau_{tgt}]) > 0 \\ 1, & \text{otherwise} \end{cases}$$

The best value of the bias $b$ is determined through cross-validation. The quantities $H$ and $H_{max}$ are respectively the entropy of the prediction and the maximum entropy of predictions over the symbol space $[\tau_{tgt}]$, and are defined as

$$H(p) = -\sum_{s \in [\tau_{tgt}]} p(s) \log_2 p(s) \,. \tag{3}$$

$$H_{max}(p) = \log_2 |S|.$$

where $p(s \in [\tau_{tgt}]) = p(\chi = s)$ is the probability mass function of a random variable $\chi$ distributed over the discrete alphabet $[\tau_{tgt}]$ such that the individual probabilities are independent and sum to 1.

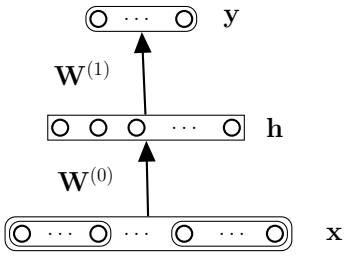
The second combination method — *product-of-experts*, is computed similarly as the weighted geometric mean of the probability distributions. This is given by

$$p(s) = \frac{1}{R} \left( \prod_{m \in M} p_m(s)^{w_m} \right)^{\frac{1}{\sum_{m \in M} w_m}}$$

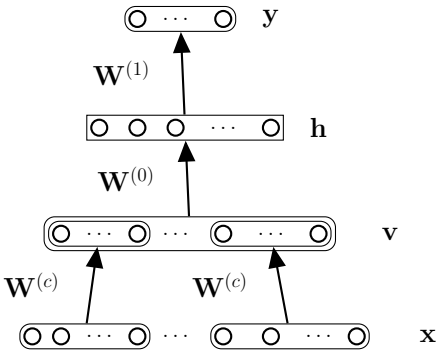
where $R$ is a normalisation constant which ensures that the resulting distribution over $S$ sums to unity. The weights $w_m$ in this case are obtained in the same manner as for the mixture-of-experts case. It was observed in a previous application of these two combination methods to melody modelling [15], that product-of-experts resulted in a greater improvement in predictive performance.

## 3. FIXED-CONTEXT NEURAL NETWORKS

In this section, we provide a brief overview of the two fixed-context neural network models which we employed for the task of predicting the pitch of the next note in a melody, given a viewpoint type context which leads up to it. These are (1) a feed-forward neural network, and (2) a neural probabilistic melody model. The key difference between the two is the presence of an additional weight-sharing layer in the latter which transforms the binary representation of the viewpoint types into lower-dimensional real-valued vectors before passing these on as inputs to a feed-forward network (much like the former).




(a) Feed-forward Neural Network




(b) Neural Probabilistic Melody Model

**Figure 1**: The two models employed for multiple viewpoint melodic prediction in this paper (biases ignored in the illustration). A concatenation of the fixed-length input type context is presented to each model in its visible layer and the predictions are made in the output layer.

### 3.1 Feed-forward Neural Network

In its simplest form, a feed-forward neural network (Figure 1) consists of an input layer $\mathbf{x} \in \mathbb{R}^n$, a hidden layer $\mathbf{h} \in \mathbb{R}^m$ and an output layer $\mathbf{y} \in \mathbb{R}^l$. The input layer is connected to the hidden layer by a weight-matrix $W^{(0)}$ and likewise, the hidden layer to the output layer by a matrix $W^{(1)}$. Each unit in the hidden layer typically applies a non-linear function to the input it receives from the layer below it. Similarly, each unit of the output layer applies a function to the input it receives from the hidden layer immediately preceding it. In a network with a single hidden layer, this happens according to the following equations

$$\mathbf{u}^{(0)} = \mathbf{b}^{(0)} + W^{(0)}\mathbf{x} \tag{4}$$
$$\mathbf{h} = f^{(0)}(\mathbf{u}) \tag{5}$$
$$\mathbf{u}^{(1)} = \mathbf{b}^{(1)} + W^{(1)}\mathbf{h} \tag{6}$$
$$\mathbf{y} = f^{(1)}(\mathbf{v}) \tag{7}$$

where $\mathbf{b}^{(0)}$ and $\mathbf{b}^{(1)}$ are the hidden and output layer biases, $f^{(0)}$ and $f^{(1)}$ are functions applied to the input received by each node in the hidden and output layers respectively. Thus, for a given input $\mathbf{x}$, the output $\mathbf{y}$ is calculated as

$$\mathbf{y} = f^{(1)}(\mathbf{b}^{(1)} + W^{(1)} \cdot f^{(0)}(\mathbf{b}^{(0)} + W^{(0)}\mathbf{x})) \tag{8}$$

In the present case, $f^{(0)}$ is the logistic sigmoid function and $f^{(1)}$ is the softmax function. The network can

be trained in a supervised manner using the backpropagation algorithm [20]. This algorithm applies the chain rule of differentiation to propagate the error between the target output and the output of the model backwards into the network, and use these derivatives to appropriately update the model parameters (the network weights and biases).

### 3.2 Neural Probabilistic Melody Model

Next we consider the neural probabilistic melody model (NPMM), which was originally introduced in [2] as a language model for word sequences. It consists of a feed-forward network such as the one described in Section 3.1, with an additional *embedding* layer below it (Figure 1). This model takes as input a concatenation of binary viewpoint type vectors (*cf.* Section 3) which represent a fixed-length context. The first layer of the network maps each of these sparse binary vectors to lower-dimensional dense real-valued vectors which make up the input layer of what is essentially a feed-forward network above it. This mapping is determined by a shared weight matrix $W^{(c)}$ which is learned from data, and is given by

$$\mathbf{v} = W^{(c)}\mathbf{x}. \tag{9}$$

The hidden layer in the case of the NPMM consists of hyperbolic-tangent activation units. The output layer contains softmax units. The model is trained with backpropagation using gradient descent as in the case of a standard feed-forward neural network.

## 4. EVALUATION

The first goal of this paper is to demonstrate the suitability of fixed-context neural networks for multiple viewpoint melodic prediction. To this end, we compare the two models described in Section 3 with variable-order Markov models (VOMMs) and restricted Boltzmann machines (RBMs). It was observed that the predictive performance of each of the neural network models is either comparable to or better than that of the best VOMMs of both bounded and unbounded order [16], while slightly worse than the RBM of [4] (Figure 2). Second, we wish to compare the predictions of a single neural network which uses multiple input types with that of an ensemble of networks with smaller input dimensions, each of which uses a subset of the input types of the former, and combined with the entropy-based weighting scheme described in 2.1. We found that, while the addition of viewpoint types does improve the predictive performance in both cases, that of the single network is slightly worse than the ensemble (Figure 3). Moreover, the extent of this improvement diminishes with an increase in context length.

### 4.1 Dataset

Evaluation was carried out on a corpus of monophonic MIDI melodies that cover a range of musical styles. It consists of 4 datasets - Bach chorale melodies, and folk melodies from Canada, China and Germany, with a total of $37,229$ musical events. These were also used to evaluate RBMs and variable order Markov models for music prediction in [4, 16]. To facilitate a direct comparison, the melodies are not transposed to a default key.

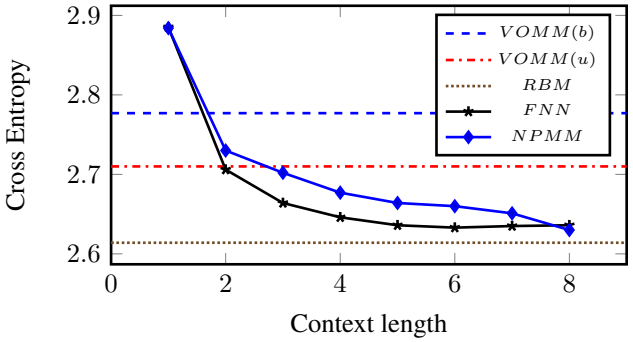



**Figure 2**: Comparison between the predictive performances of the best bounded and unbounded variable-order Markov models (VOMM(b) and VOMM(u) respectively), the best restricted Boltzmann machine (RBM), the feed-forward neural network (FNN) and the neural probabilistic melody model (NPMM) averaged over the datasets.

### 4.2 Evaluation Measure

In order to evaluate the proposed prediction models, we turn to a previous study of Markov models for music prediction in [16]. There, *cross entropy* was used to measure the information content of the models. This is a quantity related to *entropy* (3). The value of entropy, with reference to a prediction model, is a measure of the uncertainty of its predictions. A higher value reflects greater uncertainty. In practice, one rarely knows the true probability distribution of the stochastic process and uses a model to approximate the probabilities in (3). An estimate of the goodness of this approximation can be measured using cross entropy ($H_c$) which represents the divergence between the entropy calculated from the estimated probabilities and the source model. This quantity can be computed over all the subsequences of length $n$ in the test data $\mathcal{D}_{test}$, as

$$H_c(p_{mod}, \mathcal{D}_{test}) = \frac{-\sum_{s_1^n \in \mathcal{D}_{test}} \log_2 p_{mod}(s_n | s_1^{(n-1)})}{|\mathcal{D}_{test}|} \tag{10}$$

where $p_{mod}$ is the probability assigned by the model to the last pitch in the subsequence given its preceding context. Cross-entropy approaches the true entropy as the number of test samples ($|\mathcal{D}_{test}|$) increases.

### 4.3 Model Selection

Different neural network configurations were evaluated by a grid search over the learning rate $\eta = \{0.05, 0.1\}$, the number of hidden units $n_{hid} = \{25, 50, 100, 200, 400\}$, number of embedding units $n_{emb} = \{10, 20\}$ (only for the NPMM), and weight decay $w_{decay} = \{0.0000, 0.0001, 0.0005\}$. Each model was trained using mini-batch gradient descent up to a maximum of 1000 epochs with a batch size of 100 samples. Early-stopping [19] and weight-decay

were also incorporated to counter overfitting. The momentum parameter $\mu$, was set to 0.5 during the first five epochs and then increased to 0.9 for the rest of the training. Each model was evaluated with 10-fold cross-validation, with folds identical to those used in [4, 16] for the sake of comparison.

### 4.4 Model Comparison

We carried out a comparison between the predictive performance of the two neural network models presented here, and models previously evaluated on the same datasets [4, 16]. It is to be noted that, since neither of our models is updated online during prediction, the comparison with the variable order Markov models of [16] is limited to their best performing Long-Term Models. These are of order bound 2 and unbounded order (labelled there as *C*I*). It is evident from Figure 2 that both the neural network models are able to take advantage of information in longer contexts than the bounded order $n$-gram models. This is also a feature of the RBM, whose best case of context-length 5 outperforms the rest of the models in the plot. The slight deterioration in the performance of the feed-forward network for longer contexts is possibly due to poor optimization of its parameters. This is considering the fact that weight-decay and early-stopping were implemented in the training algorithm to prevent overfitting. While it was not possible to incorporate further steps for better parameter optimization in this paper, the results are still illustrative of the networks' suitability at the given task and the improvement in performance with context consistent with each other and with that of the RBMs. Possible optimizations have been left as future work, and will be discussed in Section 5.

### 4.5 Model Combination

In order to evaluate the combination of viewpoint types, we selected one type which is related to the "what" in music — scale-degree ($intfref$), and another which is related to the "when" — inter-onset interval ($ioi$), from the several possible choices that exist. Furthermore, this experiment was performed using the NPMM and only on the Chinese folk melody dataset for the purpose of illustration, with the assumption that a similar trend would be observed with the other model and datasets. As our target viewpoint type i.e. the one being predicted, is musical pitch ($seqpitch$), the first model has the input types $seqpitch$ and $intfref$ and the second one $seqpitch$ and $ioi$. The additional viewpoints are incorporated as explained in Section 2. The predictions of these two models are combined *explicitly* using the mixture- and product-of-experts schemes. On the other hand, the *implicit* combination of these two is a single model whose input types are $seqpitch$, $intfref$ and $ioi$. Figure 3 compares the predictions of the pitch-only version of the NPMM and the three models using the additional input types. It can be seen that each of these three models has a better predictive performance than its pitch-only counterpart, thus confirming the relevance of the added viewpoint types to musical pitch prediction. Both the mixture- and product-of-experts combination schemes ($seqpitch \times (intfref +_m ioi)$ and $seqpitch \times (intfref +_p ioi)$ respectively in the plot) result in very similar predictive performance, with the latter working only slightly better for shorter context-lengths of 1, 2 and 3. Moreover, both these explicit combinations of viewpoint types perform better than the single implicit combination of types ($seqpitch \times intfref \times ioi$ in the plot). One will, however, notice that the cross entropy of the predictions slightly worsens at longer context-lengths, and that the discrepancy between the implicit and explicit combinations gradually increases in these cases. As mentioned earlier, we attribute this to the optimization of the network parameters, which is to be dealt with in future work.

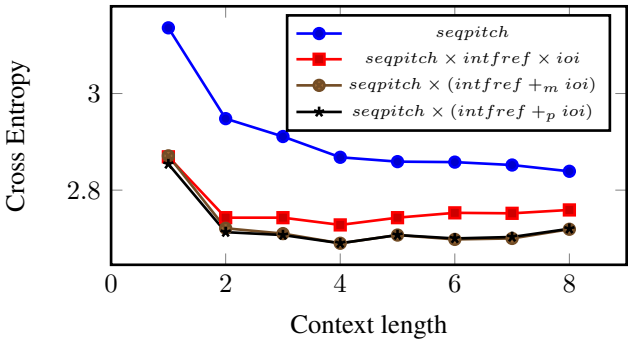



**Figure 3**: Comparison between the predictive performances, on the Chinese folk melody dataset, of the pitch-only NPMM, its extension which uses the *intfref* and *ioi* types as additional input, and ensembles each of which combines two models of input types (a) *seqpitch* and *intfref* (b) *seqpitch* and *ioi* using the mixture ($+_m$) and product ($+_p$) combination schemes.

## 5. CONCLUSIONS & FUTURE WORK

The two neural network models for melodic prediction presented here have been found to have a predictive performance comparable to or better than previously evaluated VOMMs, but slightly worse than that of RBMs. Predictive performance can be further improved by the addition of viewpoint types to the same model, or by combining multiple models using an entropy-weighted combination scheme. In our experiments, the latter tended to be better.

One open issue that remains is the parameter optimization in the two networks presented here. It was observed that, particularly when the input layer of a network is large and the dataset relatively small, the predictive performance does not improve as expected with context-length and the addition of viewpoint types. We note here that the results presented have been generated with models implemented in-house [1] for use with the Python machine learning library *scikit-learn* [18], and were thus limited in the various initialization and optimization strategies used in their learning algorithms. We also suspect this to be the reason for the limited success of the NPMM which exhibited relatively more promising results in its language

[1] Code available upon request.

modelling application in [2]. Many more measures to improve generalization and overall prediction accuracy (such as dropout, different weights initialization strategies and layer-wise pre-training) have been suggested in [1]. Incorporating these measures (or using an existing neural network library which does) can further improve the results.

Apart from this, there are three other aspects which are of immediate interest to us. The first is the incorporation of a short-term element in the prediction model which updates its parameters as data is presented to it, and has been shown to result in improved prediction performance and human-like predictions [15]. Secondly, while the number of parameters of the fixed-context models presented here increases linearly with the context-length (assuming a fixed number of hidden units), we are at present experimenting with recurrent networks where this problem does not arise due to their recurrent connections. And finally, the extension of the said models to polyphonic multiple viewpoints representations is also an open issue at the moment which we hope to address in the future.

## 6. ACKNOWLEDGEMENTS

Srikanth Cherla is supported by a PhD studentship from City University London. The authors would like to thank Marcus Pearce for his valuable advice, and the anonymous reviewers for their feedback on the submission.

## 7. REFERENCES

[1] Yoshua Bengio. Practical Recommendations for Gradient-Based Training of Deep Architectures. In *Neural Networks: Tricks of the Trade*, pages 437–478. 2012.

[2] Yoshua Bengio, Rejean Ducharme, Pascal Vincent, and Christian Jauvin. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3:1137–1155, 2003.

[3] Srikanth Cherla, Artur d'Avila Garcez, and Tillman Weyde. A neural probabilistic model for predicting melodic sequences. In *International Workshop on Machine Learning and Music*, 2013.

[4] Srikanth Cherla, Tillman Weyde, Artur d'Avila Garcez, and Marcus Pearce. A distributed model for multiple viewpoint melodic prediction. In *International Society for Music Information Retrieval Conference*, pages 15–20, 2013.

[5] Darrell Conklin. Prediction and entropy of music. 1990.

[6] Darrell Conklin. Music generation from statistical models. In *AISB Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, pages 30–35, 2003.

[7] Darrell Conklin. Multiple viewpoint systems for music classification. *Journal of New Music Research*, 42(1):19–26, 2013.

[8] Darrell Conklin and John G Cleary. Modelling and generating music using multiple viewpoints. 1988.

[9] Darrell Conklin and Ian H Witten. Multiple viewpoint systems for music prediction. *Journal of New Music Research*, 24(1):51–73, 1995.

[10] Greg Cox. On the relationship between entropy and meaning in music: An exploration with recurrent neural networks. *Proceedings of the 32nd Annual Cognitive Science Society. Austin TX: CSS*, 2010.

[11] David Brian Huron. *Sweet anticipation: Music and the psychology of expectation*. MIT press, 2006.

[12] Christopher D Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

[13] Leonard B Meyer. Meaning in music and information theory. *The Journal of Aesthetics and Art Criticism*, 15(4):412–424, 1957.

[14] Diana Omigie, Marcus T Pearce, Victoria J Williamson, and Lauren Stewart. Electrophysiological correlates of melodic processing in congenital amusia. *Neuropsychologia*, 2013.

[15] Marcus T Pearce. *The construction and evaluation of statistical models of melodic structure in music perception and composition*. PhD thesis, City University London, 2005.

[16] Marcus T Pearce and Geraint A Wiggins. Improved methods for statistical modelling of monophonic music. *Journal of New Music Research*, 33(4):367–385, 2004.

[17] Marcus T Pearce and Geraint A Wiggins. Expectation in melody: The influence of context and learning. *Music Perception*, 23(5):377–405, June 2006.

[18] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.

[19] Lutz Prechelt. Early Stopping But When? In *Neural Networks: Tricks of the Trade*, pages 55–69. 2012.

[20] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 1988.

[21] Raymond P Whorley. *The Construction and Evaluation of Statistical Models of Melody and Harmony*. PhD thesis, 2013.